Package 'HZIP'

October 23, 2026

Type Package

Title Likelihood-Based Inference for Joint Modeling of Correlated Count and Binary Outcomes with Extra Variability and Zeros

Version 0.1.5

Author Lizandra C. Fabio [aut],
Jalmar M. F. Carrasco [aut, cre],
Victor H. Lachos [aut],
Ming-Hui Chen [aut]

Maintainer Jalmar M. F. Carrasco <carrasco.jalmar@ufba.br>

Description Inference approach for jointly modeling correlated count and binary outcomes. This formulation allows simultaneous modeling of zero inflation via the Bernoulli component while providing a more accurate assessment of the Hierarchical Zero-Inflated Poisson's parsimony (Lizandra C. Fabio, Jalmar M. F. Carrasco, Victor H. Lachos and Ming-Hui Chen, Likelihood-based inference for joint modeling of correlated count and binary outcomes with extra variability and zeros, 2026, under submission).

License GPL-3

Encoding UTF-8

LazyData true

LinkingTo Rcpp, RcppParallel, RcppArmadillo

Imports Rcpp, Formula, pscl, stats, tibble, dplyr, statmod, RcppParallel, ggplot2, numDeriv, rlang

Depends R (>= 3.5)

URL https://github.com/carrascojalmar/HZIP

BugReports https://github.com/carrascojalmar/HZIP/issues

Config/roxygen2/version 8.1.0

RoxygenNote 7.3.3

NeedsCompilation yes

Repository CRAN

Date/Publication 2026-10-23 18:40:02 UTC

Contents

envelope 2
hzip 3
residuals.hzip 6
rhzip 8
salamanders 10
testDisp 11
testZI 13

Index **15**

`envelope` *Envelope Plot for HZIP Models*

Description

Produces a normal Q-Q plot with a simulation envelope for randomized quantile residuals from a Hierarchical Zero-Inflated Poisson (HZIP) model fitted via `hzip`.

Usage

```
envelope(object, nsim = 21, method = "ghq", Q = 15, ...)
```

Arguments

`object`	Either an object of class `hzip` (typically obtained from `hzip`), or a numeric vector of randomized quantile residuals obtained from `residuals.hzip`.
`nsim`	Integer specifying the number of Monte Carlo simulations used to construct the simulation envelope. Default is `21`.
`method`	Character string passed to `residuals.hzip` when `object` is of class `hzip`. One of `"ghq"` (default) or `"laplace"`. Ignored if `object` is a numeric vector.
`Q`	Integer giving the number of Gauss-Hermite quadrature nodes when `method = "ghq"`. Default is `15`. Ignored if `object` is a numeric vector or `method = "laplace"`.
`...`	Additional arguments (currently ignored).

Details

The envelope is constructed using Monte Carlo simulation based on `nsim` replications of independent standard normal samples. For each order statistic, the 2.5% and 97.5% empirical quantiles define the envelope limits.

If `object` is of class `hzip`, residuals are computed internally via `residuals.hzip` before plotting.

Value

A `ggplot2` object containing the Q-Q plot with simulation envelope.

See Also

hzip, residuals.hzip

Examples

```
fit.salamander <- hzip(
  y ~ mined+spp+mined:spp | mined+spp+sample,
  data = salamanders
)

# Passing the fitted object directly
envelope(fit.salamander)

# Or passing residuals explicitly
res <- residuals(fit.salamander)
envelope(res)
```

---

hzip *Fit a Hierarchical Zero-Inflated Poisson (HZIP) Model*

Description

`hzip()` fits a longitudinal/clustered zero-inflated Poisson model with subject-level random effects following a Generalized Log-Gamma (GLG) distribution, by maximizing a marginal likelihood approximated via adaptive Gauss-Hermite quadrature. The model uses a two-part Formula: $y \sim$ zero part | count part, where the count intensity (Poisson mean) and the zero-inflation probability are linked to (possibly different) sets of covariates. Initial values are obtained from `pscl::zeroinfl(..., dist = "poisson", link = "cloglog")`.

Usage

```
hzip(
  formula,
  data,
  hessian = TRUE,
  method = "BFGS",
  Q = 15,
  lower = -Inf,
  upper = Inf,
  control = NULL,
  start = NULL,
  refine = FALSE,
  ep_method = c("optim", "numDeriv"),
  n_starts = 1,
  verbose = FALSE,
  ...
)
```

Arguments

formula	A two-part Formula of the form `y ~ x_zero + ... | w_count + ...` (matching the column-naming convention used by rhzip: the zero-inflation design comes from `x1`, the count design from `x2`), where the right-hand side before the bar specifies covariates for the zero-inflation component and the right-hand side after the bar specifies covariates for the Poisson mean.
data	A `data.frame` containing all variables used in `formula` and a subject identifier named `Ind` (one row per observation).
hessian	Logical; if `TRUE` (default) the observed Hessian at the optimum is computed and used for standard-error estimates.
method	Character string passed to optim for the first optimization stage (default `"BFGS"`).
Q	Integer; number of Gauss-Hermite nodes for quadrature (default `15`). Larger values improve accuracy at higher computational cost.
lower	Bounds on the variables for the `"L-BFGS-B"` method (passed to optim).
upper	Bounds on the variables for the `"L-BFGS-B"` method (passed to optim).
control	Optional `list` passed to optim's `control=` argument (e.g., `list(maxit = 500)`).
start	Optional numeric vector of initial values of length $p_1 + p_2 + 2$ in the order `c(scale_zero, beta_zero, scale_count, beta_count)`. If `NULL` (default), initial values are obtained from `pscl::zeroinfl` and the scales are set to `0.5`.
refine	Logical; if `TRUE`, the BFGS optimum is further refined by nlminb (a trust-region Newton method). Useful as a safeguard for difficult problems but typically does not improve the solution when BFGS already converges. Default `FALSE` (BFGS only).
ep_method	Character; method used to compute the Hessian for the standard errors. Either `"optim"` (default; uses the finite-difference Hessian computed by optim, fast) or `"numDeriv"` (uses hessian with Richardson extrapolation, more accurate but slower). For most applications the two give virtually identical standard errors; `"numDeriv"` is recommended only when extra accuracy is needed.
n_starts	Integer; if greater than `1`, performs a multi-start procedure with `n_starts` perturbations of the initial values and keeps the best log-likelihood. Useful for difficult problems with possible local optima. Default `1` (no multi-start).
verbose	Logical; if `TRUE`, prints progress information during the optimization. Default `FALSE`.
...	Further arguments passed to optim.

Details

Let y_{ij} denote the count response for subject i at occasion j. The HZIP model assumes

$$P(y_{ij} = 0 \mid u_i) = \pi_{ij}(u_i) + \{1 - \pi_{ij}(u_i)\} \exp\{-\mu_{ij}(u_i)\},$$

$$P(y_{ij} = k \mid u_i) = \{1 - \pi_{ij}(u_i)\} \frac{\mu_{ij}(u_i)^k e^{-\mu_{ij}(u_i)}}{k!}, \quad k \geq 1,$$

with linear predictors for the count and zero parts (links typically `log` for the Poisson mean and `cloglog` for the zero-inflation). Subject-specific random effects u_i follow a GLG distribution and

induce within-subject dependence; the marginal likelihood is approximated by adaptive Gauss-Hermite quadrature with `Q` nodes.

Optimization strategy. The default workflow is:

1. Generate initial values from `pscl::zeroinfl` (with scales set to a neutral `0.5`, ensuring reproducibility);
2. Run `optim` with the specified `method` (default BFGS);
3. If `refine = TRUE`, refine the result with `nlminb` (off by default; useful for difficult problems);
4. If `n_starts > 1`, repeat steps 2-3 with perturbed initial values and keep the best log-likelihood;
5. Compute standard errors from the inverse Hessian using the method specified by `ep_method`.

Value

An object of class `"hzip"`, a `list` with elements:

`call`	The matched call.
`formula`	The model `Formula`.
`coefficients_zero`	Estimated coefficients for the zero-inflation part.
`coefficients_count`	Estimated coefficients for the count part.
`scale_zero`	Estimated scale (zero part).
`scale_count`	Estimated scale (count part).
`loglik`	Maximized log-likelihood.
`AIC`, `BIC`	Information criteria based on `loglik`.
`convergence`	Convergence code (0 = success).

(`loglikz`, `AICz`, `BICz` and the `AIC_HZIP`/`BIC_HZIP` contrasts are currently disabled/commented out in the code and are not part of the returned object or printed output for now.)

`n`	Number of observations.
`m`	Cluster sizes per subject (vector ordered by `Ind`).
`ep`	Approximate standard errors (square roots of the diagonal of the inverse Hessian).
`iter`	Number of optimization iterations.
`method`	Optimization method.
`refine`	Whether `nlminb` refinement was applied.
`ep_method`	Method used to compute standard errors.
`n_starts`	Number of starting points used.
`optim`	Raw output from the final optimization stage.
`data`	The input `data`.

Note

The subject identifier must be named `Ind`. The internal parameter vector follows the order `c(scale_zero, beta_zero, scale_count, beta_count)`.

For difficult problems (e.g., models with many interaction terms), if the default fit produces large standard errors or the AIC suggests a worse fit than a nested simpler model, try (i) `n_starts = 10` for multi-start optimization and (ii) supplying `start = ...` from a fit of the simpler nested model.

References

Min, Y., & Agresti, A. (2005). Random effect models for repeated measures of zero-inflated count data. *Statistical Modelling*, 5(1), 1-19.

Jackman, S. (2020). *pscl*: Classes and Methods for R Developed in the Political Science Computational Laboratory. R package version 1.5.5.

Zeileis, A., & Croissant, Y. (2010). Extended model formulas in R: *Journal of Statistical Software*, 34(1), 1-13. (**Formula**)

Examples

```
fit.salamander <- hzip(y ~ mined+spp+mined:spp | mined+spp+sample,
data = salamanders)
summary(fit.salamander)
```

`residuals.hzip` *Residuals for HZIP Models*

Description

Computes randomized quantile residuals for fitted `hzip` models.

Usage

```
## S3 method for class 'hzip'
residuals(object, method = c("ghq", "laplace"), Q = 21, ...)
```

Arguments

`object`	An object of class `hzip`, typically obtained from `hzip`.
`method`	Character string indicating the approximation method used to estimate the posterior random effects. Possible values are: `"ghq"` Gauss-Hermite quadrature approximation. `"laplace"` Laplace approximation based on posterior modes.
`Q`	Integer specifying the number of Gauss-Hermite quadrature nodes when `method = "ghq"`. Ignored when `method = "laplace"`.
`...`	Further arguments passed to or from other methods.

Details

The residuals are obtained by integrating out the latent random effects using either Gauss-Hermite quadrature or a Laplace approximation.

Let Y_{ij} denote the response variable for the j-th observation of the i-th subject. The residuals are computed using the randomized quantile residual approach proposed by Dunn and Smyth (1996).

For each observation, the conditional cumulative distribution function is evaluated as

$$F^*(y_{ij}) = F(y_{ij} - 1) + U_{ij} P(Y_{ij} = y_{ij}),$$

where $U_{ij} \sim \text{Uniform}(0, 1)$.

The residuals are then obtained as

$$r_{ij} = \Phi^{-1}(F^*(y_{ij})),$$

where $\Phi^{-1}(\cdot)$ is the standard normal quantile function.

Random effects are approximated through either:

- Gauss-Hermite quadrature (`method = "ghq"`);
- Laplace approximation (`method = "laplace"`).

The Laplace approximation computes posterior modes using `optim`, while the quadrature approach relies on `gauss.quad`.

Value

A numeric vector containing the randomized quantile residuals.

References

Dunn, P. K. and Smyth, G. K. (1996). Randomized quantile residuals. *Journal of Computational and Graphical Statistics*, 5(3), 236–244.

Examples

```
fit.salamander <- hzip(y ~ mined+spp+mined:spp | mined+spp+sample,
  data = salamanders)

res1 <- residuals(fit.salamander, method = "ghq")
res2 <- residuals(fit.salamander, method = "laplace")

head(res1)
head(res2)
```

rhzip *Simulate Data from a Hierarchical Zero-Inflated Poisson (HZIP) Model*

Description

`rhzip()` generates panel/longitudinal data from a two–part hierarchical zero-inflated Poisson model with subject-specific random effects for both the zero-inflation and the count components. Random effects are drawn from a generalized log-gamma (GLG) distribution via `rgengamma()` (user must provide/attach a function with this name; see **Dependencies**).

Usage

```
rhzip(n, m, para, x1, x2)
```

Arguments

`n`	Integer. Number of subjects.
`m`	Integer vector of length `n` (or a scalar recycled to length `n`). Numbers of repeated measurements per subject.
`para`	Numeric vector of parameters in the order `c(lambda1, beta1, lambda2, beta2)`, where: • `lambda1`: GLG scale/shape parameter for the zero part. • `beta1`: length-`p1` vector of coefficients for the zero part (matches `ncol(x1)`). • `lambda2`: GLG scale/shape parameter for the count part. • `beta2`: length-`p2` vector of coefficients for the count part (matches `ncol(x2)`). Internally, the linear predictors are $\eta_{ij}^{(0)} = x_{1,ij}^{\top}\beta_1 + b_i^{(0)}$ and $\eta_{ij}^{(1)} = x_{2,ij}^{\top}\beta_2 + b_i^{(1)}$, with $p_{ij} = 1 - \exp\{-\exp(\eta_{ij}^{(0)})\}$ (cloglog link for zero-inflation) and $\mu_{ij} = \exp(\eta_{ij}^{(1)})$ (log link for counts).
`x1`	Numeric matrix of covariates for the zero-inflation part (dimension `sum(m)` by `p1`). Include an intercept column if desired; every column supplied is kept in the output (see **Column naming**).
`x2`	Numeric matrix of covariates for the count part (dimension `sum(m)` by `p2`). Include an intercept column if desired; every column supplied is kept in the output (see **Column naming**).

Details

For subject $i = 1, \ldots, n$ with m_i observations, the model draws two subject-level random effects $b_i^{(0)}$ and $b_i^{(1)}$ independently from GLG distributions parameterized by `lambda1` and `lambda2`. Conditional on these effects, outcomes are generated as

$$Y_{ij} = Z_{ij} \times C_{ij},$$

where $Z_{ij} \sim \text{Bernoulli}(p_{ij})$ controls structural zeros and $C_{ij} \sim \text{Poisson}(\mu_{ij})$ controls the count size.

The returned data frame contains subject IDs, the response `y`, and the supplied covariates. An attribute `"propZeros"` stores a small summary with the percentage of structural zeros, additional Poisson zeros, and total zeros.

Value

A `tibble` with columns:

`Ind`	Subject identifier (1..n), repeated according to `m`.
`y`	Simulated response.
`x*`, `w*`	The covariates from `x1` and `x2` (renamed as described below).

The object has an attribute `"propZeros"`: a 3×1 `data.frame` with rows `"Zeros"` (structural zeros, %), `"Count"` (extra zeros from the Poisson part, %), and `"Total"` (overall zero percentage).

Column naming

All columns of `x1` are kept and renamed `x1, x2, ..., xp1` (zero-inflation part). All columns of `x2` are kept and renamed `w1, w2, ..., wp2` (count part). No column is dropped: if `x1` and/or `x2` include an intercept column (a column of 1's), it is preserved in the output (as `x1` and/or `w1`, respectively) and can be referenced explicitly in the `hzip` formula (using `-1` on that side to avoid R adding a duplicate automatic intercept), or simply omitted from the formula (letting R's automatic intercept play the same role).

Dependencies

This function calls `rgengamma()` to draw GLG random effects. Ensure that such a function is available on the search path (e.g., from a package that provides a generalized log-gamma RNG) or provide your own implementation with the signature `rgengamma(n, mu, sigma, lambda)`. It also uses **dplyr** and **tibble**.

See Also

`hzip` for model fitting.

Examples

```
set.seed(123)

n <- 50
m <- rep(4, n)
N <- sum(m)

# design matrices (with intercepts)
x1 <- cbind(1, rnorm(N))
x2 <- cbind(1, rnorm(N), rbinom(N, 1, 0.5))

p1 <- ncol(x1); p2 <- ncol(x2)
```

```
lambda1 <- 0.7
beta1   <- c(-0.2, 0.6)
lambda2 <- 0.9
beta2   <- c( 0.3, 0.5, -0.4)

para <- c(lambda1, beta1, lambda2, beta2)

sim <- rhzip(n, m, para, x1, x2)
head(sim)
attr(sim, "propZeros")

# Fit it back with hzip(). With the new convention NO column is dropped,
# so x1 (2 cols: intercept, slope) comes out as "x1,x2" and x2 (3 cols:
# intercept, slope, slope) comes out as "w1,w2,w3". Referencing only the
# real slopes ("x2" and "w2+w3") and letting R's formula add the
# intercept automatically reproduces the true model exactly (equivalently,
# one could write "x1+x2-1 | w1+w2+w3-1" using the explicit intercept
# columns and disabling the automatic one).

fit <- hzip(y ~ x2 | w2 + w3, data = sim)
summary(fit)


# Example WITHOUT an intercept in the zero part (e.g. simulation Scenario 1
# style): x1 has no column of 1's, so every column of x1 is a real slope
# and must be fit with "-1" to avoid an automatic (spurious) intercept.

n2 <- 200; m2 <- rep(4, n2); N2 <- sum(m2)
w1b <- rep(rnorm(n2), times = m2)
w2b <- rep(runif(n2), times = m2)
x1b <- cbind(w1b, w2b)             # zero part: NO intercept
x2b <- cbind(1, w1b, w2b)          # count part: WITH intercept
parab <- c(0.8, 1, -1, 0.5, 1, 1, 1)  # (lambda1,beta11,beta12,lambda2,beta20,beta21,beta22)
simb <- rhzip(n2, m2, parab, x1b, x2b)
head(simb)  # columns: Ind, y, x1, x2 (zero part, both real slopes), w1, w2, w3 (count part)
fitb <- hzip(y ~ x1 + x2 - 1 | w2 + w3, data = simb)
summary(fitb)
```

`salamanders` *Salamanders data*

Description

This dataset is adapted from the **glmmTMB** package and contains salamander counts with information on mining status and species. It is intended for illustrating zero-inflated Poisson models with random effects using the `hzip()` function.

Usage

```
salamanders
```

Format

A data frame with 644 rows and 5 variables:

Ind Cluster identifier (integer, 1 to 161); each cluster groups the 4 repeated samples of one site-species combination.

y Count response variable (integer).

mined Mining status: `"yes"` or `"no"`.

spp Species factor with multiple levels (e.g., GP, PR, DM, etc.).

sample Repeated sampling occasion (factor with levels `"1"` to `"4"`).

Details

The dataset was originally included in the **glmmTMB** package (Brooks et al., 2017), and has been slightly modified for testing the **HZIP** package: observations are ordered by site, the variables `count`, `mined`, `spp` and `sample` are kept, `count` is renamed to `y`, and the cluster identifier `Ind` is added.

Source

Adapted from the **glmmTMB** package.

Examples

```
data(salamanders, package = "HZIP")

## Fit zero-inflated Poisson with random effects
fit.salamander <- hzip(y ~ mined+spp+mined:spp | mined+spp+sample,
                       data = salamanders)
summary(fit.salamander)
```

---

testDisp *Simulation-Based Dispersion Test for HZIP Models*

Description

Compares the observed variance of the response with the distribution of variances obtained from parametric bootstrap replicates simulated under the fitted `hzip` model.

Usage

```
testDisp(
  object,
  nsim = 500,
  alternative = c("two.sided", "greater", "less"),
  alpha = 0.05,
  seed = NULL,
  plot = TRUE,
  ...
)
```

Arguments

`object`	An object of class `hzip`.
`nsim`	A positive integer giving the number of simulated replicates used to build the reference distribution. Defaults to `500`.
`alternative`	Character string specifying the alternative hypothesis. One of `"two.sided"` (default), `"greater"` (overdispersion), or `"less"` (underdispersion).
`alpha`	Numeric value in `(0, 1)` giving the significance level used for the decision. Defaults to `0.05`.
`seed`	An optional integer seed for reproducibility.
`plot`	Logical; if `TRUE` (default) a histogram of the simulated statistics is printed with the observed value marked.
`...`	Further arguments (currently ignored).

Value

An object of class `"hzip_test"` (invisibly), a list with components:

`statistic` Observed variance of the response.

`p.value` Simulation-based p-value.

`alternative` The alternative hypothesis used.

`alpha` Significance level used for the decision.

`simulated` Numeric vector of simulated variances.

`method` Description of the test.

`h0` Statement of the null hypothesis.

The result is printed automatically; to change the number of significant digits shown, use `print(res, digits = 6)`, where `res` is the object returned by `testDisp()`.

See Also

`testZI`, `hzip`

Examples

```
fit.salamander <- hzip(
  y ~ mined+spp+mined:spp | mined+spp+sample,
  data = salamanders
)

testDisp(fit.salamander, nsim = 200, seed = 42)
```

`testZI` *Simulation-Based Zero-Inflation Test for HZIP Models*

Description

Compares the observed proportion of zeros with the distribution of zero proportions obtained from parametric bootstrap replicates simulated under the fitted `hzip` model.

Usage

```
testZI(
  object,
  nsim = 500,
  alternative = c("two.sided", "greater", "less"),
  alpha = 0.05,
  seed = NULL,
  plot = TRUE,
  ...
)
```

Arguments

`object`	An object of class `hzip`.
`nsim`	A positive integer giving the number of simulated replicates. Defaults to `500`.
`alternative`	Character string specifying the alternative hypothesis. One of `"two.sided"` (default), `"greater"` (more zeros than expected), or `"less"` (fewer zeros than expected).
`alpha`	Numeric value in `(0, 1)` giving the significance level used for the decision. Defaults to `0.05`.
`seed`	An optional integer seed for reproducibility.
`plot`	Logical; if `TRUE` (default) a histogram of the simulated statistics is printed with the observed value marked.
`...`	Further arguments (currently ignored).

Value

An object of class "hzip_test" (invisibly), a list with components:

`statistic` Observed proportion of zeros.

`p.value` Simulation-based p-value.

`alternative` The alternative hypothesis used.

`alpha` Significance level used for the decision.

`simulated` Numeric vector of simulated zero proportions.

`method` Description of the test.

`h0` Statement of the null hypothesis.

The result is printed automatically; to change the number of significant digits shown, use `print(res, digits = 6)`, where `res` is the object returned by `testZI()`.

See Also

`testDisp`, `hzip`

Examples

```
fit.salamander <- hzip(
  y ~ mined+spp+mined:spp | mined+spp+sample,
  data = salamanders
)

testZI(fit.salamander, nsim = 200, seed = 42)
```

Index

∗ **datasets**
salamanders, 10

envelope, 2

Formula, *3, 4*

gauss.quad, *7*

hessian, *4*
hzip, *2, 3*, 3, *6*, *9*, *12*, *14*

nlminb, *4, 5*

optim, *4, 5*, *7*

residuals.hzip, *2, 3*, 6
rhzip, *4*, 8

salamanders, 10

testDisp, 11, *14*
testZI, *12*, 13