

Package ‘HZIP’

May 7, 2026

Type Package

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

Version 0.1.1

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, 2025, under submission).

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

LinkingTo Rcpp, RcppParallel, RcppArmadillo

Imports Rcpp, Formula, pscl, stats, tibble, dplyr, statmod,
RcppParallel, cubature, VGAM, ggplot2

Depends R (>= 3.5)

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

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

NeedsCompilation yes

Repository CRAN

Date/Publication 2025-12-19 17:10:15 UTC

Contents

envelope.HZIP	2
hzip	3
residuals.HZIP	5
rHZIP	6
salamanders	8

Index	10
--------------	-----------

envelope.HZIP	<i>Envelope simulation for HZIP Model</i>
---------------	---

Description

Produces a Q-Q plot of residuals from a Hierarchical Zero-Inflated Poisson (HZIP) Model fitted via [hzip](#).

Usage

```
envelope.HZIP(object, nsim = 100, ...)
```

Arguments

object	An object of class HZIP, typically returned by hzip .
nsim	Integer. Number of simulations used to construct the envelope. Default is 100.
...	Additional arguments (currently ignored).

Details

A simulation envelope is added using Monte Carlo replications.

Value

Envelope simulation plot.

See Also

[hzip](#), [residuals.HZIP](#)

Examples

```
data(salamanders)
fit.salamander <- hzip(y ~ mined|mined+spp,data = salamanders)
res <- residuals(fit.salamander)
envelope.HZIP(res, nsim = 21)
```

Description

`hzip()` fits a longitudinal/clustered zero-inflated Poisson model with subject-level random effects by maximizing a (marginal) likelihood approximated. The model uses a two-part [Formula](#): y 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,
  ...
)
```

Arguments

formula	A two-part Formula of the form $y \sim w_zero + \dots \mid x_count + \dots$, 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 <code>data.frame</code> containing all variables used in <code>formula</code> and a subject identifier named <code>Ind</code> (one row per observation).
hessian	Logical; if <code>TRUE</code> (default) the observed Hessian at the optimum is returned and used to compute standard-error estimates.
method	Character string passed to <code>optim</code> (default <code>"BFGS"</code>).
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, or bounds in which to search for method "Brent" (arguments passed to <code>optim</code>).
upper	method, or bounds in which to search for method "Brent" (arguments passed to <code>optim</code>).
control	Optional list passed to <code>optim</code> 's <code>control</code> =argument (e.g., <code>list(maxit = 500)</code>).
...	Further arguments passed to <code>optim</code> .

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

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	Optimized objective value returned by <code>optim</code> . (Note: depending on <code>lvero</code> , this may be the negative log-likelihood.)
convergence	<code>optim</code> convergence code.
n	Number of observations or subjects (see Note).
m	Cluster sizes per subject (vector ordered by <code>Ind</code>).
ep	Approximate standard errors (square roots of the diagonal of the inverse Hessian).
iter	Number of <code>optim</code> iterations.
method	Optimization method.
optim	Raw <code>optim</code> output.
data	The input data.

Note

The subject identifier must be named `Ind`. The sign convention for the zero-part coefficients in the initial values follows `pscl::zeroinfl`; the internal parameter vector is `c(scale_zero, -beta_zero, scale_count, beta_count)`. Also verify whether `loglik` is the (negative) log-likelihood as returned by your objective `lvero`; if it is the negative log-likelihood, you may want to store `logLik = -op$value` for user convenience.

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|mined+spp,data = salamanders)
summary(fit.salamander)
```

residuals.HZIP *Compute Residuals for HZIP Models*

Description

This function calculates residuals for objects of class HZIP using randomized quantile residuals. The computation is performed efficiently using C++ functions for predicting random effects and calculating residuals.

Usage

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

Arguments

`object` An object of class HZIP, typically returned from [hzip](#).

`...` Additional arguments (not used).

Details

The function internally groups the data by individual (Ind), constructs model matrices for both zero-inflation and count parts of the model, and then calls the C++ functions `predict_HZIP_cpp_vec` and `r_ij_cpp_vec` to efficiently compute the residuals. Random effects are integrated using adaptive quadrature based on the supplied nodes and weights.

Value

A numeric vector of residuals with length equal to the total number of observations in the dataset.

Examples

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

rHZIP	<i>Simulate Data from a Hierarchical Zero-Inflated Poisson (HZIP) Model</i>
-------	---

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: <ul style="list-style-type: none"> • 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. |
| x2 | Numeric matrix of covariates for the count part (dimension sum(m) by p2). Include an intercept column if desired. |

Details

For subject $i = 1, \dots, n$ 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:

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

The object has an attribute `"propZeros"`: a 3x1 data frame with rows `"Zeros"` (structural zeros, %), `"Count"` (extra zeros from the Poisson part, %), and `"Total"` (overall zero percentage).

Column naming

Columns of `x1` are renamed to `x1`, `x2`, ..., `xp1`. Columns of `x2` are copied except the first column is dropped and the remaining are renamed `w1`, `w2`, ..., `w_{p2-1}`. (This mirrors the current implementation that excludes the first `x2` column from the output.)

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")
```

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 4 variables:

Ind Individual identifier (factor).

y Count response variable (integer).

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

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

Details

The dataset was originally included in the **glmmTMB** package (Brooks et al., 2017), and has been slightly modified for testing the **HZIP** package.

Source

Adapted from the **glmmTMB** package.

Examples

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

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

Index

* **datasets**

salamanders, [8](#)

envelope.HZIP, [2](#)

Formula, [3](#)

hzip, [2](#), [3](#), [5](#), [7](#)

optim, [3](#)

residuals.HZIP, [2](#), [5](#)

rHZIP, [6](#)

salamanders, [8](#)