

Package ‘HZIP’

June 2, 2026

Type Package

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

Version 0.1.2

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

RoxygenNote 7.3.2

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>

NeedsCompilation yes

Repository CRAN

Date/Publication 2026-06-02 20:20:02 UTC

Contents

envelope.HZIP	2
hzip	3
print.hzip_test	6
residuals.HZIP	7
rHZIP	8
salamanders	10
simulate.HZIP	11
testDisp	13
testDisp.HZIP	13
testZI	14
testZI.HZIP	15
Index	17

envelope.HZIP	<i>Envelope Plot for HZIP Models</i>
---------------	--------------------------------------

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.HZIP(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 100.
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 21. 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 | mined + spp,  
  data = salamanders  
)  
  
# Passing the fitted object directly  
envelope.HZIP(fit.salamander)  
  
# Or passing residuals explicitly  
res <- residuals(fit.salamander)  
envelope.HZIP(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 $y \sim w_zero + \dots \mid x_count + \dots$, 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 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 <code>pscl::zeroinfl</code> 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

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

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

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 `nlmminb` (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:

<code>call</code>	The matched call.
<code>formula</code>	The model Formula.
<code>coefficients_zero</code>	Estimated coefficients for the zero-inflation part.
<code>coefficients_count</code>	Estimated coefficients for the count part.
<code>scale_zero</code>	Estimated scale (zero part).
<code>scale_count</code>	Estimated scale (count part).
<code>loglik</code>	Maximized log-likelihood.
<code>loglikz</code>	Auxiliary log-likelihood from <code>mlez_hat</code> .
AIC, BIC	Information criteria based on <code>loglik</code> .
AICz, BICz	Information criteria based on <code>loglikz</code> .
<code>convergence</code>	Convergence code (0 = success).
<code>n</code>	Number of observations.
<code>m</code>	Cluster sizes per subject (vector ordered by <code>Ind</code>).
<code>ep</code>	Approximate standard errors (square roots of the diagonal of the inverse Hessian).
<code>iter</code>	Number of optimization iterations.
<code>method</code>	Optimization method.
<code>refine</code>	Whether <code>nlmminb</code> refinement was applied.
<code>ep_method</code>	Method used to compute standard errors.
<code>n_starts</code>	Number of starting points used.
<code>optim</code>	Raw output from the final optimization stage.
<code>data</code>	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

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

# More robust fit for difficult problems (multi-start)
fit.salamander2 <- hzip(y ~ mined | mined + spp + mined:spp,
  data = salamanders,
  n_starts = 10, verbose = TRUE)
summary(fit.salamander2)

# Using the simpler model as starting point for the harder one
init <- with(fit.salamander, c(scale_zero, coefficients_zero,
  scale_count, coefficients_count,
  rep(0, 6))) # 6 zeros for the interaction
fit.salamander3 <- hzip(y ~ mined | mined + spp + mined:spp,
  data = salamanders, start = init)
summary(fit.salamander3)
```

print.hzip_test

Print Method for hzip_test Objects

Description

Prints a formatted summary of a simulation-based test returned by `testDisp.HZIP` or `testZI.HZIP`, including the null hypothesis, observed statistic, p-value, and decision.

Usage

```
## S3 method for class 'hzip_test'
print(x, digits = 4, ...)
```

Arguments

x An object of class hzip_test.
 digits Number of significant digits to print. Default is 4.
 ... Further arguments (currently ignored).

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 | mined + spp,
  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: <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:

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 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`, \dots , `xp1`. Columns of `x2` are copied except the first column is dropped and the remaining are renamed `w1`, `w2`, \dots , `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)
```

simulate.HZIP

Simulate Responses from a Fitted HZIP Model

Description

Generates `nsim` replicated response vectors from a fitted HZIP model by sampling latent random effects from the Generalized Log-Gamma distribution and then drawing from the hurdle-Poisson component.

Usage

```
## S3 method for class 'HZIP'
simulate(object, nsim = 1, seed = NULL, ...)
```

Arguments

object	An object of class HZIP, typically obtained from hzip .
nsim	A positive integer giving the number of response vectors to simulate. Defaults to 1.
seed	An optional integer seed passed to set.seed for reproducibility. If NULL (default) the current random state is used.
...	Further arguments (currently ignored).

Details

For each replicate and each observation i , the algorithm:

1. Samples $b_{1i} \sim \text{LGG}(0, \sigma_1, \lambda_1)$ and $b_{2i} \sim \text{LGG}(0, \sigma_2, \lambda_2)$ independently via `rgengamma()`.
2. Computes $\pi_i = 1 - \exp\{-\exp(\eta_i + b_{1i})\}$ and $\mu_i = \exp(\rho_i + b_{2i})$.
3. Draws Y_i : with probability π_i sets $Y_i = 0$; otherwise draws from a zero-truncated Poisson(μ_i) via rejection sampling.

Value

A data.frame with n rows and $nsim$ columns, where n is the number of observations in object. Each column is one simulated response vector, named `sim_1`, `sim_2`, ..., `sim_nsim`. This follows the convention of the S3 generic [simulate](#).

See Also

[hzip](#), [testDisp.HZIP](#), [testZI.HZIP](#)

Examples

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

sims <- simulate(fit.salamander, nsim = 10, seed = 42)
dim(sims) # n x 10
head(sims)
```

testDisp	<i>Test for Overdispersion</i>
----------	--------------------------------

Description

Generic function for simulation-based dispersion tests.

Usage

```
testDisp(object, ...)
```

Arguments

object	A fitted model object.
...	Further arguments passed to methods.

testDisp.HZIP	<i>Simulation-Based Dispersion Test for HZIP Models</i>
---------------	---------------------------------------------------------

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

```
## S3 method for class 'HZIP'
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.

See Also

[testZI.HZIP](#), [simulate.HZIP](#)

Examples

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

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

testZI

Test for Zero-Inflation

Description

Generic function for simulation-based zero-inflation tests.

Usage

```
testZI(object, ...)
```

Arguments

object	A fitted model object.
...	Further arguments passed to methods.

testZI.HZIP

*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

```
## S3 method for class 'HZIP'
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.

See Also

[testDisp.HZIP](#), [simulate.HZIP](#)

Examples

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

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

Index

- * **datasets**
 - salamanders, 10
- envelope.HZIP, 2
- Formula, 3, 4
- gauss.quad, 8
- hessian, 4
- hzip, 2, 3, 3, 7, 10, 12
- nllminb, 4, 5
- optim, 4, 5, 8
- print.hzip_test, 6
- residuals.HZIP, 2, 3, 7
- rHZIP, 8
- salamanders, 10
- set.seed, 12
- simulate, 12
- simulate.HZIP, 11, 14, 16
- testDisp, 13
- testDisp.HZIP, 6, 12, 13, 16
- testZI, 14
- testZI.HZIP, 6, 12, 14, 15