

Package: ZIHINAR1 (via r-universe)

May 19, 2026

Type Package

Title Zero-Inflated and Hurdle INAR(1) Models

Version 0.1.0

Maintainer Fusheng Yang <fusheng.yang@uconn.edu>

Description Provides tools for estimating Zero-Inflated INAR(1) (ZI-INAR(1)) and Hurdle INAR(1) (H-INAR(1)) models using 'Stan'. It allows users to simulate time series data for these models, estimate parameters, and evaluate model fit using various criteria. Functions include model estimation, simulation, and likelihood-based metrics.

License GPL-3

Encoding UTF-8

LazyData true

Imports rstan, VGAM, actuar, matrixStats, coda, ggplot2, gridExtra, stats, knitr, utils

Suggests testthat (>= 3.0.0), rmarkdown

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Depends R (>= 4.0.0)

Config/pak/sysreqs make

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

Date/Publication 2025-11-19 15:19:07 UTC

RemoteUrl <https://github.com/fushengyy/zihinar1>

RemoteRef HEAD

RemoteSha 2fc7135638adfa4d7f3c1f71e1e50c303989bb15

Contents

data_simu	2
get_est	3

get_mod_sel	4
get_pred	5
get_stanfit	5

Index	7
--------------	----------

data_simu	<i>Simulate Time Series Data for ZI-INAR(1) or H-INAR(1) Models</i>
-----------	---

Description

This function simulates time series data for Zero-Inflated INAR(1) (ZI-INAR(1)) or Hurdle INAR(1) (H-INAR(1)) models, using either Poisson or Negative Binomial distributions.

Usage

```
data_simu(n, alpha, rho, theta, mod_type, distri)
```

Arguments

n	Integer specifying the number of observations to simulate.
alpha	Numeric value between 0 and 1 representing the autoregressive parameter.
rho	Numeric value between 0 and 1 representing the zero-inflation or hurdle parameter.
theta	Numeric vector of model parameters. For Poisson, it should be <code>theta[1] = lambda</code> . For Negative Binomial, it should be <code>theta = c(lambda, phi)</code> , where <code>phi</code> is the size parameter.
mod_type	Character string indicating the model type. Use "zi" for zero-inflated models and "h" for hurdle models.
distri	Character string specifying the distribution. Options are "poi" for Poisson or "nb" for Negative Binomial.

Value

A numeric vector containing the simulated time series data.

References

Part of the implementation of this function was adapted from the **ZINAR1** package. The **ZINAR1** package simulates first-order integer-valued autoregressive processes with zero-inflated innovations (ZINAR(1)) and estimates its parameters under a frequentist approach.

For more information about the ZINAR1 package, please refer to:

Aldo M. Garay, João Vitor Ribeiro (2022). *ZINAR1: Simulates ZINAR(1) Model and Estimates Its Parameters Under Frequentist Approach*. R package version 0.1.0. Available at: <https://CRAN.R-project.org/package=ZINAR1>.

Garay, A. M., Ribeiro, J. V. (2021). First-Order Integer Valued AR Processes with Zero-Inflated Innovations. In: *Nonstationary Systems: Theory and Applications*, Springer. DOI: [doi:10.1007/9783030821104_2](https://doi.org/10.1007/9783030821104_2).

We acknowledge the original authors, Aldo M. Garay and João Vitor Ribeiro, for their contributions.

Examples

```
# Simulate 50 observations from a Zero-Inflated Poisson INAR(1) model
y_data <- data_simu(n = 50, alpha = 0.5, rho = 0.3, theta = c(5),
                  mod_type = "zi", distri = "poi")
head(y_data)
```

get_est

Get Parameter Estimates from Stan Model Fit

Description

Extracts parameter estimates from a Stan model fit, including mean, median, standard deviation, and HPD intervals.

Usage

```
get_est(distri, stan_fit)
```

Arguments

`distri` Character string specifying the distribution. Options are "poi" or "nb".
`stan_fit` A stanfit object returned by `get_stanfit`.

Value

A summary of the parameter estimates.

Examples

```
# Generate toy data
y_data <- data_simu(n = 60, alpha = 0.5, rho = 0.3, theta = c(5),
                  mod_type = "zi", distri = "poi")

# Fit a small Stan model (may take > 5s on first compile)
stan_fit <- get_stanfit(mod_type = "zi", distri = "poi", y = y_data)

# Get parameter estimates from the Stan model fit
get_est(distri = "poi", stan_fit = stan_fit)
```

`get_mod_sel`*Get Model Selection Criteria*

Description

Calculates model selection criteria such as AIC, BIC, DIC, and WAIC from a Stan model fit.

Usage

```
get_mod_sel(y, mod_type, distri, stan_fit)
```

Arguments

<code>y</code>	A numeric vector representing the observed data.
<code>mod_type</code>	Character string indicating the model type ("zi" or "h").
<code>distri</code>	Character string specifying the distribution ("poi" or "nb").
<code>stan_fit</code>	A stanfit object returned by <code>get_stanfit</code> .

Value

A summary table of model selection criteria, including:

EAIC Expected Akaike Information Criterion (AIC).

EBIC Expected Bayesian Information Criterion (BIC).

DIC Deviance Information Criterion (DIC).

WAIC1 First version of Watanabe-Akaike Information Criterion (WAIC).

WAIC2 Second version of Watanabe-Akaike Information Criterion (WAIC).

The summary is printed in a table format for easy interpretation.

Examples

```
# Generate toy data
y_data <- data_simu(n = 60, alpha = 0.5, rho = 0.3, theta = c(5),
  mod_type = "zi", distri = "poi")

# Fit a small Stan model (may take > 5s on first compile)
stan_fit <- get_stanfit(mod_type = "zi", distri = "poi", y = y_data)

# Get model selection criteria
get_mod_sel(y = y_data, mod_type = "zi", distri = "poi",
  stan_fit = stan_fit)
```

get_pred	<i>Get Predictions from Stan Model Fit</i>
----------	--

Description

Extracts predicted values from a Stan model fit.

Usage

```
get_pred(stan_fit)
```

Arguments

stan_fit A stanfit object returned by get_stanfit.

Value

A summary of the predictions and bar charts of each prediction.

Examples

```
# Generate toy data
y_data <- data_simu(n = 60, alpha = 0.5, rho = 0.3, theta = c(5),
                  mod_type = "zi", distri = "poi")

# Fit a small Stan model (may take > 5s on first compile)
stan_fit <- get_stanfit(mod_type = "zi", distri = "poi", y = y_data)

# Get predicted values from the Stan model fit
get_pred(stan_fit = stan_fit)
```

get_stanfit	<i>Fit ZI-INAR(1) or H-INAR(1) Model using Stan</i>
-------------	---

Description

This function fits a Zero-Inflated INAR(1) (ZI-INAR(1)) or Hurdle INAR(1) (H-INAR(1)) model using Stan and returns the model fit.

Usage

```
get_stanfit(  
  mod_type,  
  distri,  
  y,  
  n_pred = 4,  
  thin = 2,  
  chains = 1,  
  iter = 2000,  
  warmup = iter/2,  
  seed = NA  
)
```

Arguments

mod_type	Character string indicating the model type. Use "zi" for zero-inflated models and "h" for hurdle models.
distri	Character string specifying the distribution. Options are "poi" for Poisson or "nb" for Negative Binomial.
y	A numeric vector of integers representing the observed data.
n_pred	Integer specifying the number of time points for future predictions (default is 4).
thin	Integer indicating the thinning interval for Stan sampling (default is 2).
chains	Integer specifying the number of Markov chains to run (default is 1).
iter	Integer specifying the total number of iterations per chain (default is 2000).
warmup	Integer specifying the number of warmup iterations per chain (default is iter/2).
seed	Numeric seed for reproducibility (default is NA).

Value

A stanfit object containing the Stan model fit.

Examples

```
# Generate toy data  
y_data <- data_simu(n = 60, alpha = 0.5, rho = 0.3, theta = c(5),  
  mod_type = "zi", distri = "poi")  
  
# Fit the model using Stan (small config)  
stan_fit <- get_stanfit(mod_type = "zi", distri = "poi", y = y_data)  
print(stan_fit)
```

Index

data_simu, 2
get_est, 3
get_mod_sel, 4
get_pred, 5
get_stanfit, 5