Journal of Behavioral Data Science, 2022, *2* (2), 156–173.
DOI: `https://doi.org/10.35566/jbds/v2n2/shao`

A Tutorial on Bayesian Analysis of Count Data Using JAGS

Sijing (SJ) Shao[1]

Department of Psychology, University of Notre Dame, Notre Dame, USA
`sshao2@nd.edu`

Abstract. In behavioral studies, the frequency of a particular behavior or event is often collected and the acquired data are referred to as count data. This tutorial introduces readers to Poisson regression models which is a more appropriate approach for such data. Meanwhile, count data with excessive zeros often occur in behavioral studies and models such as zero-inflated or hurdle models can be employed for handling zero-inflation in the count data. In this tutorial, we aim to cover the necessary fundamentals for these methods and equip readers with application tools of `JAGS`. Examples of the implementation of the models in `JAGS` from within `R` are provided for demonstration purposes.

Keywords: Count data · Zero-inflation · Poisson regression · ZIP model · Hurdle model

1 Introduction

In behavioral studies, information such as the number of times a certain behavior or event occurs is often collected in order to help understand individuals. Such collected nonnegative and discrete data are typically called count data. While normal distribution is the commonly used distribution in most research, specifying a normal distribution for such outcome variables can be inappropriate for at least two reasons: (1) negative and real expected values in a normal distribution is possible while only nonnegative integer values are allowed in such count data; and (2) the distribution of count data is often positively skewed and its variance usually increases along with its mean, while mean and variance are assumed to be unrelated in normal distributions. Poisson regression is more appropriate than general linear regression for such count data, and it models the non-negative integer responses against linear predictors through a link function.

Meanwhile, count data in behavioral research are often heavily skewed due to large amount of zero responses. The zero responses consist of responses from either the individuals who never engaged in such behaviors or those who have engaged but not currently (Grimm & Stegmann, 2019). For example, in alcohol use disorder (AUD) studies, the number of alcohol drinks is often collected from the participants. A zero response could either from participants who never engaged in drinking behavior, or those who drink but not when they are being sampled. The zero-inflated Poisson (ZIP; Lambert (1992)) model was proposed when zeros are assumed to be from both scenarios while the hurdle model (Mullahy, 1986) is appropriate when zeros are assumed to be from only one source. In this tutorial, Poisson regression models for count data, as well as ZIP and hurdle models for zero-inflated response variables are discussed under the Bayesian framework. Examples of estimating these models with `JAGS` (Plummer, 2003) and `R` (Team, 2013) package `runjags` (Denwood, 2016) are illustrated.

1.1 Poisson Regression

The responses $\mathbf{Y} = (Y_1, .., Y_n)^T$ are count of independent events occur in a fixed time interval for n participants. The likelihood function for each response is specified as:

$$Y_i \sim Poisson(\lambda_i), \lambda_i > 0,$$

where the rate parameter λ_i denotes the average number of count per time interval for each person. The density function can be written as $p(Y_i = k) = \frac{e^{-\lambda_i}\lambda_i^k}{k!}$ for $k > 0$. The parameter λ_i can be modeled as a linear function of a set of predictors X with a log link function such that: $log(\lambda_i) = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + ... + \beta_P X_{Pi}$. The parameter β_0 is the intercept, which is the predicted mean of $exp(Y)$ when X is 0. The parameter β_j is the coefficient corresponding to the changes in predictor X_p: one unit increase in X is associated with the expected change in the outcome $exp(Y)$.

1.2 Zero-inflated Poisson (ZIP) Model

In the ZIP framework, the excess zero observations are from either individuals who never engaged in the behaviors of interest, with probability p_i, or individuals who are part of the Poisson distribution in which zeros are generated from participants who have engaged in the behavior but not when the survey was conducted, with probability $1 - p_i$:

$$Y_i \sim \begin{cases} 0, & \text{with probability } p_i \\ Poisson(\lambda_i), & \text{with probability } 1 - p_i, \end{cases}$$

where i indicates the ith participant. The parameter λ_i is the mean parameter for Poisson distribution and represents the expected event frequency for individual i. Thus, $p(Y_i = 0) = p_i + (1 - p_i) \times \frac{e^{-\lambda_i}\lambda_i^0}{0!} = p_i + (1 - p_i) \times e^{-\lambda_i}$ and $p(Y_i = k) = (1 - p_i)\frac{e^{-\lambda_i}\lambda_i^k}{k!}$ for $k > 0$. Let X be the covariates that affect the Poisson mean and B be the covariates affect the probability p_i through log and $logit$ link functions respectively:

$$\begin{aligned} log(\lambda_i) &= \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + ... + \beta_P X_{Pi} \\ logit(p_i) &= \gamma_0 + \gamma_1 B_{1i} + \gamma_2 B_{2i} + ... + \gamma_J B_{Ji}, \end{aligned}$$

where βs and γs are coefficients for design matrices, which include a column of 1 as intercept and predictors X or B respectively. The interpretation for β_0 and β_1 is similar as in Poisson regression. γ_0 and γ_1 are specific to ZIP model. When B is zero, the average odds for a participant to be in the "zero only" group vs. "Poisson" group is $exp(\gamma_0)$. With one unit increases in B, the odds that a participant would be in the "zero only" group vs. "Poisson" group increases by a factor of $exp(\gamma_1)$.

1.3 Hurdle Model

While there are two types of individuals in ZIP, the hurdle model treats all participants in the same way so that everyone could be engaged in the behavior when the survey was undertaken: they could decide to be engaged in the behavior, and then the intensity of the behavior. Thus, two processes are involved in the hurdle model. For n independent observations Y_i:

$$Y_i \sim \begin{cases} 0, & \text{with probability } p_i \\ \text{truncated Poisson}(\lambda_i) & \text{with probability } 1 - p_i. \end{cases}$$

In contrast to ZIP which includes logistic regression to predict "excess zeros" over and above the zeros predicted by Poisson, hurdle models uses logistic regression to predict zero vs non-zeros. The "hurdle" is used to measure whether a response falls below or above the hurdle (e.g., the hurdle is zero in this case). The positive responses above the hurdle zero are then modeled by other truncated count regressions. In this framework, $p(Y_i = 0) = p_i$ and $p(Y_i = k) = \frac{(1-p_i)(\lambda_i^k e^{-\lambda_i})}{(1-e^{-\lambda_i})k!}$ for $k > 0$. Similar to the ZIP framework, design matrices, which include a column of 1 as intercept and predictors X or B, are associated with $log(\lambda_i)$ and $logit(p_i)$ through coefficients βs and γs respectively. However, the interpretation for γs is slightly different from in ZIP: When B is zero, the average odds for a participant not engaging vs. engaging in the behavior is $exp(\gamma_0)$. With one unit increase in B, the odds that a participant would not be engaging vs. engaging in the behavior increases by a factor of $exp(\gamma_1)$.

2 Model Estimation

2.1 Data Description

The data on number of recreational boating trips to Lake Somerville was collected in 1980. The dataset includes 659 responses from registered leisure boat owners in 23 counties in Texas. Figure 1 reveals its variability from 0 up to around 80 with large amount of zero responses. It clearly suggests that this count variable is not normally distributed. The number of recreational trips is often associated with the annual household income. In this illustrative example, we examine whether income is a predictor of number of recreational boating trips a person took. The income variable measures the annual household income of the respondent (in 1,000 USD) and is centered at its mean for the purpose of interpretability. The data is available in R package AER (Kleiber & Zeileis, 2008).

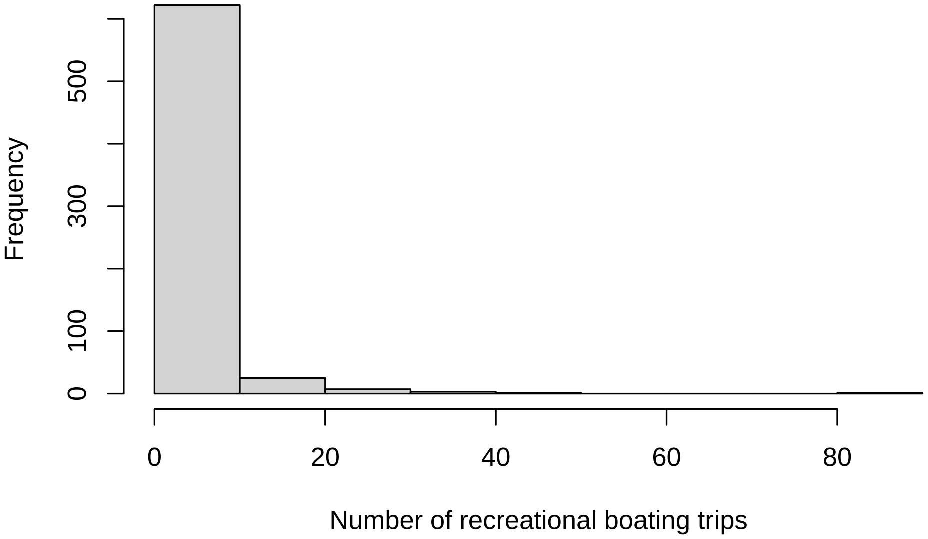

Figure 1. Histogram of the outcome variable.

2.2 Estimation of Poisson Regression in `runjags`

We consider a model in which $log(\lambda)$ is a linear function of income, where λ_i denotes the average number of vacations person i took. The regression equation is:

$$log(\lambda_i) = \beta_0 + \beta_1 \times \text{incomeC}.$$

The model can be estimated in R with package `runjags` and the function `run.jags` is used. It requires a valid model definition, string of monitored variables (`beta0` for β_0 and `beta1` for β_1 in this example), and data, as discussed below.

```
Pois_Est <- run.jags(model = Poisson_Model, monitor =
  c("beta0",  "beta1", "exp_beta0", "exp_beta1"),
  data = data, n.chains = 3,  inits = inits,
  method = "simple", adapt = 1000, burnin = 3000,
  sample = 10000)
```

Some of the important arguments used in the function `run.jags` are:

- model. The model can be specified as a character string, including the likelihood function and initial values. In order to estimate the Poisson regression model using `JAGS`, we first need to specify its likelihood function for all participants and define λ_i with the log link function (see Lines 2 - 6 below). Note the operator “ $\sim$” is used to define random variables and it represents “is distributed as”. Line 6 `Y[i] ~ dpois(lambda[i])` means that the response `Y[i]` is distributed as a Poisson distribution with rate parameter `lambda[i]`. The operator <- is for the linear function: `log(lambda_i) <- beta0 + beta1*X[i]`. Non-informative priors for β_0 and β_1 can be set as β_0, $\beta_1 \sim N(0, 1000)$. In `JAGS`, a normal distribution is specified as `dnorm(mu, tau)`, with mean `mu` and precision `tau`, where precision is the reciprocal of the variance. Thus, $N(0, 1000)$ is specified as `dnorm(0, 1/1000)` in `JAGS`, see Lines 9 - 10.

```
 1 Poisson_Model <- "model{
 2    ## Likelihood ##
 3    for (i in 1:N){
 4     Y[i] ~ dpois(lambda[i])
 5     log(lambda[i]) <- beta0 + beta1*X[i]
 6    }
 7
 8    ## priors for coefficients
 9     beta0 ~ dnorm(0, 1/1000)
10     beta1 ~ dnorm(0, 1/1000)
11
12    ## exponentiate the paramters
13     exp_beta0 <- exp(beta0)
14     exp_beta1 <- exp(beta1)
15 }"
```

- monitor. The parameters to be estimated are defined in a character string. Since the coefficient parameters are in log odds scale in Poisson regressions, their exponentiated parameters should be obtained for interpretation. In`JAGS`, the exponentiated parameters `exp_beta0` for $exp(\beta_0)$ and `exp_beta1` for $exp(\beta_1)$ can be sampled directly. `exp_beta0` and `exp_beta1` need to specified in `monitor` argument as well as in `model`.
- inits. We need to prepare initial values for `beta0` and `beta1` as shown below. A set of initial values is specified as a list regarding the parameters to be estimated. When multiple chains are generated for convergence diagnosis, a nested list using the `inits` argument with length equal to the number of chains should be specified. In this example, three sets of initial values are specified since three chains are used for convergence diagnosis.

```
inits <- list(list(beta0 = rnorm(1, 0, 0.1),
                   beta1 = rnorm(1, 0, 0.1)),
              list(beta0 = 1, beta1 = 1),
              list(beta0 = -1, beta1 = -1))
```

- data. The variables from data are specified as a list and passed into the data used in `JAGS`, with the argument `data`. The outcome variable Y in `Poisson_Model` is the "trips" variable from the raw data `dat`, denoted as `Y = dat$trips`; the predictor X is the centered "income" variable, denoted as `X = dat$incomeC`. In `Poisson_Model`, `N` is the total sample size and need to be defined as `N = nrow(dat)`.

```
data <- list(Y = dat$trips, X = dat$incomeC,
             N = nrow(dat))
```

- n.chains. Multiple chains can be generated for convergence diagnostic. In this example, three chains were simulated and denoted as `n.chain = 3`. More chains will cause the simulation to run more slowly.
- method. A number of simulation methods are provided in `JAGS`. `simple` is specified here since the model in the illustration example is relatively simple. When more simulation time is possible, other methods allowing parallelisation should be considered.
- adapt. A adaption process is often needed for MCMC samplers in `JAGS` to sample the posteriors more efficiently. The default is 1000 iterations.
- burnin. MCMC samplers often take a finite number of iterations to find the region of posterior probability and this portion of chains should be discarded for inference. The default is 4000 iterations. `burnin` and `adapt` should be specified separately.
- sample. The total number of MCMC samples for each chain can be specified. The default is 10,000 iterations.

Convergence Diagnosis Three Markov chains are obtained with three different set of initial values. The traceplots of the Markov chains for β_0 and β_1 and their exponential values are in Figure 2. Since there is no clear trend in either of the plots and three chains are mixed well, it suggests that convergence is achieved. In addition, the potential scale reduction factor (psrf; (Gelman & Rubin, 1992)) in Table 1 are close to 1, suggesting again that convergence has been reached.

Interpretation The exponentiated coefficients along with the HPD intervals for β_0 and β_1 are shown in Table 1. When the household income is at the average level (3,853 USD), the expected number of boat trips took by the respondents is approximately 2 (2.21) times.

Table 1. Bayesian parameter estimates from Poisson regression

	Mean	SD	Lower 95	Upper 95	psrf
beta0	0.79	0.03	0.74	0.84	1.0002
beta1	-0.10	0.02	-0.13	-0.07	1.0004
exp_beta0	2.21	0.06	2.09	2.32	1.0002
exp_beta1	0.90	0.01	0.88	0.93	1.0004

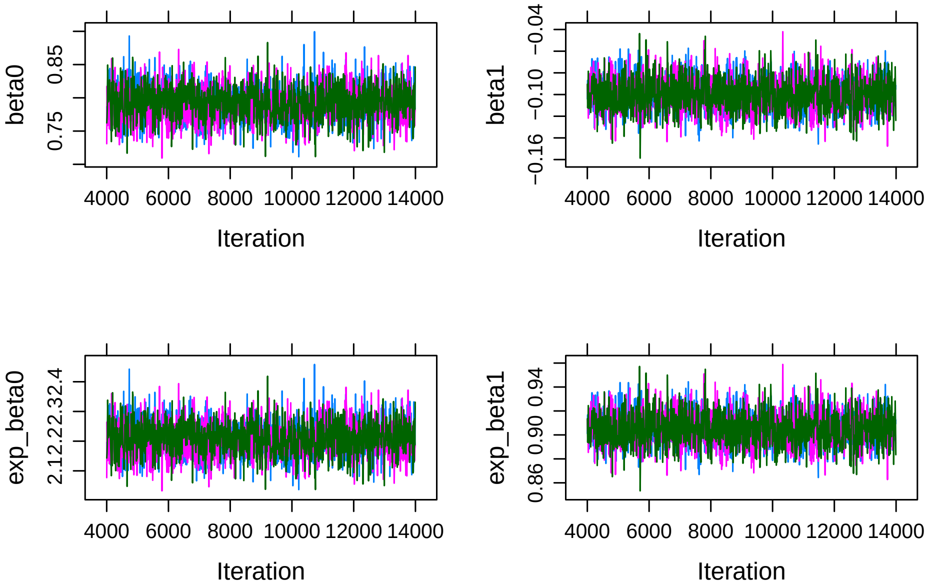

Figure 2. Traceplots of beta0 and beta1 from Poisson regression.

In Table 1, “lower 95” is the 2.5 percentile of the HPD interval and “upper 95” is the 95 percentile. Since 1 is not included in the HPD interval for $exp(\beta_1)$ [0.88, 0.93], the predictor income is statistically associated with number of boat trips took by the respondents. With every $1000 increases in the annual household income, the expected number of boat trips taken by the respondents decreases by 10% (1- (exp(-0.1)) ×100%) on average.

2.3 Estimation of ZIP in runjags

The arguments `inits`, `data`, and `model` used in function `run.jags` for the ZIP model are specified in similar ways as in Poisson regression.

```
ZIP_Est <- run.jags(model = ZIP_model,
  monitor = c("beta0", "beta1", "gamma0", "gamma1"),
  data = data, n.chains = 3, inits = inits,
  method = "simple", adapt = 1000, burnin = 3000,
  sample = 10000, keep.jags.files = T, tempdir = T)
```

Similar to estimating Poisson regression in `JAGS`, both likelihood function and prior for parameters are specified first in `ZIP_model`.The likelihood function is defined in Lines 2 - 9. In ZIP, the probability of a zero response coming from the excessive zeros, which are from the group of respondents who never took a boat trip (`W[i] = 0`), is p_i. The probability of a zero response generated from sampling zeros, who usually take boat trips but not when they are being sampled is $1 - p_i$. The sampling zeros are zeros that are generated from the Poisson distribution, denoted as `W[i] = 1`. `W` is a latent Bernoulli random variable and is related to predictor centered income with logit link function:

$$logit(p_i) = \gamma_0 + \gamma_1 \times \text{incomeC}.$$

In another word, when `W[i]` is 0, `W[i]*mu[i]` or `lambda[i]` becomes 0. `Y[i]` is generated from the excessive zeros, which is the group of respondents who never took a boat trip. When `W[i]` is 1, `Y[i]` is generated from the Poisson distribution with rate parameter `mu[i]`. The regression equation for λ_i is as same as in Poisson regression:

$$log(\lambda_i) = \beta_0 + \beta_1 \times \text{incomeC}.$$

Note that the covariates for $logit(p_i)$ and $log(\lambda_i)$ can be the same. For simplicity, we use centered income as the predictor for both. Non-informative prior $N(0, 10000)$ is specified for the four estimated parameters β_0, β_1, γ_0, and γ_1 in Lines 12 - 15.

```
1 ZIP_model <- "model{
2  ## likelihood
3  for (i in 1:N){
4   Y[i] ~ dpois(lambda[i])
5   lambda[i] <- W[i]*mu[i]
```

```
  W[i] ~ dbern(1-p[i])
  log(mu[i]) <- beta0 + beta1*X[i]
  logit(p[i]) <- gamma0 + gamma1*B[i]
 }

 ## prior
  beta0 ~ dnorm(0, 1/10000)
  beta1 ~ dnorm(0, 1/10000)
  gamma0 ~ dnorm(0, 1/10000)
  gamma1 ~ dnorm(0, 1/10000)

 ## exponentiate the paramters
  exp_beta0 <- exp(beta0)
  exp_beta1 <- exp(beta1)
  exp_gamma0 <- exp(gamma0)
  exp_gamma1 <- exp(gamma1)
}"
```

Initial values for gamma0 and gamma1 are set in the same way as beta0 and beta1. One new variable W is introduced and binary initial values of it are generated. The length of W is the total sample size for the data. See Lines 1 - 11 for details. The data specification for ZIP is as same as for Poisson regression as the same dataset is used.

```
1  W <- dat$trips
2  W[dat$trips>0] <- 1
3
4  inits <- list(list(beta0 = rnorm(1, 0, 0.1),
5                     beta1 = rnorm(1, 0, 0.1),
6                     gamma0 = rnorm(1, 0, 0.1),
7                     gamma1 = rnorm(1, 0, 0.1), W = W),
8                list(beta0 = 1, beta1 = 1, gamma0 = 1,
9                     gamma1 = 1, W = W),
10               list(beta0 = -1, beta1 = -1, gamma0 = 1,
11                    gamma1 = 1, W = W))
12
13 data <- list(Y = dat$trips, X = dat$incomeC,
14   B = dat$incomeC, N = nrow(dat))
```

Convergence Diagnosis The methods for convergence diagnosis for ZIP model is as same as for Poisson regression models. Thus, the details are omitted here.

Interpretation The exponentiated coefficients and their HPD intervals for the four estimated parameters are shown in Table 2. The results suggest that for a respondent from a household with average income, the odds of a zero response collected from this person indicating that s/he never went on a boat trip vs. s/she have taken a boat trip but not when being sampled is 1.71. This effect is significant since its HPD interval [1.45, 2] does not contains 1. In addition, when the annual household income increases by 1000 USD, the odds of a zero response being generated from these who never went on a boat trip vs. those who usually took a boat trip but not when being sampled decreases by 3% ($(1 - 0.97) \times 100\%$). This effect is not significant since its HPD interval [0.88, 1.05] contains 1.

Meanwhile, for these who have taken a boat trip but not when being sampled, an increase of $1000 in annual household income is associated with 13% ($(1 - 0.87) \times 100\%$) less average number of boat trips taken by the respondents. This effect is significant since the corresponding HPD interval [0.84, 0.9] does not contain 1.

Table 2. Bayesian parameter estimates from the ZIP model

	Mean	SD	Lower 95	Upper 95	psrf
beta0	1.78	0.03	1.73	1.84	1.0002
beta1	-0.14	0.02	-0.18	-0.11	1.0001
gamma0	0.53	0.08	0.38	0.69	1.0001
gamma1	-0.03	0.05	-0.12	0.05	1.0000
exp_beta0	5.95	0.16	5.63	6.26	1.0002
exp_beta1	0.87	0.02	0.84	0.90	1.0001
exp_gamma0	1.71	0.14	1.45	2.00	1.0001
exp_gamma1	0.97	0.04	0.88	1.05	1.0000

2.4 Estimation of Hurdle Models in `runjags`

In contrast to ZIP where both count and binary parts generate zeros, only the binary part modeled by logistic function in hurdle models generates zeros. The nonzero responses are assumed to be from a truncated Poisson distribution. Zero trick is used when setting up the Bayesian model in `runjags`. The details of zero trick approach are discussed in (Ntzoufras, 2011) and the code for the likelihood function specification is in Lines 2 - 14. `C <- 10000` is specified for the zero trick to make `-ll[i] + C` greater than 0. A dummy variable `z[i]` is created so that it is 0 when `Y[i]` is smaller than 0.0001 and is 1 otherwise.

The log likelihood of the truncated Poisson distribution `truncPois[i]` is defined in Lines 6 - 7. The total likelihood function is the sum of `z[i]*(log(1-p[i]) + truncPois)` and `(1-z[i])*log(p[i])`. When `z[i]` is 0 (`Y[i]` is 0, or `Y[i]` is from the zero-only group), the total likelihood function is `log(p[i])`; when `z[i]` is 1 (`Y[i]` is positive, or `Y[i]` is from the truncated Poisson group), the total likelihood function is `log(1-p[i]) + truncPois`.

Non-informative priors $N(0, 10000)$ is specified in Lines 18 - 21 for the four parameters β_0, β_1, γ_0, and γ_1.

```
hurdle_model <- "model{
 # likelihood
 C <- 10000
 for (i in 1:N){
  zeros[i] ~ dpois(-ll[i] + C)
  truncPois[i] <- Y[i]*log(mu[i]) - mu[i]
     - (log(1-exp(-mu[i])) + logfact(Y[i]))

  l1[i] <- (1-z[i])*log(p[i])
  l2[i] <- z[i]*(log(1-p[i]) + truncPois[i])
  ll[i] <- l1[i] + l2[i]

  log(mu[i]) <- beta0 + beta1*X[i]
  logit(p[i]) <- gamma0 + gamma1*B[i]
 }

 # prior
  beta0 ~ dnorm(0, 1/10000)
  beta1 ~ dnorm(0, 1/10000)
  gamma0 ~ dnorm(0, 1/10000)
  gamma1 ~ dnorm(0, 1/10000)
}"
```

Similar to ZIP, the initial values for the four parameters β_0, β_1, γ_0, and γ_1 are set as below. A column of zeros is added to data for the zero trick approach. Values of z[i] are generated from raw data and are provided to the argument data.

```
inits <- list(list(beta0 = rnorm(1, 0, 0.1),
                   beta1 = rnorm(1, 0, 0.1),
                   gamma0 = rnorm(1, 0, 0.1),
                   gamma1 = rnorm(1, 0, 0.1)),
              list(beta0 = 1, beta1 = 1, gamma0 = 1,
                   gamma1 = 1),
              list(beta0 = -1, beta1 = -1, gamma0 = 1,
                   gamma1 = 1))
z<-dat$trips
z[dat$trips > 0] <- 1
data <- list(Y = dat$trips, X = dat$incomeC,
             B = dat$incomeC, N = nrow(dat),
             z = z, zeros = rep(0, nrow(dat)))
```

Convergence Diagnosis The methods for convergence diagnosis for hurdle model are the same for Poisson regression models. Thus, the details are omitted here.

Interpretation The exponentiated coefficients and the corresponding HPD intervals are presented in Table 3. For a respondent from a household with average income, the odds of this person not have been to vs. have been to a boat trip is 1.73. This is significant since its HPD interval [1.47, 2.01] does not contain 1. The odds decrease by 2% ($(1 - 0.98) \times 100\%$) when the average house hold income increases by $1000. This effect is not significant since its HPD interval [0.9, 1.07] contains 1.

Meanwhile, for those who have taken a trip when being sampled, an increase of $1000 in annual household income is associated with 13% ($(1 - 0.87) \times 100\%$) less average number of boat trips taken by the respondents. This effect is significant since the corresponding HPD interval [0.84, 0.9] does not contain 1.

Table 3. Bayesian parameter estimates from the hurdle model

	Mean	SD	Lower 95	Upper 95	psrf
beta0	1.78	0.03	1.73	1.84	1.0002
beta1	-0.14	0.02	-0.18	-0.11	1.0003
gamma0	0.55	0.08	0.39	0.71	1.0003
gamma1	-0.02	0.04	-0.10	0.07	1.0002
exp_beta0	5.96	0.16	5.65	6.27	1.0002
exp_beta1	0.87	0.02	0.84	0.90	1.0003
exp_gamma0	1.73	0.14	1.47	2.01	1.0003
exp_gamma1	0.98	0.04	0.90	1.07	1.0001

3 Discussion

This tutorial covered methods handling count data and how these methods can be estimated in Bayesian framework with `runjags`. When the data is positively skewed with zero inflation, ZIP and hurdle models can be considered to handle such scenarios. Even though ZIP and hurdle models have been employed interchangeably in psychological research, they are described in two distinct frameworks: ZIP is a mixture model in which zeros can be generated from both Poisson and Bernoulli distributions while hurdle is a two-part model separating zeros from positive responses. While the output tables for ZIP and hurdle models suggest that the results are very similar, the interpretation of the models differs. Researchers should be careful with the choice of methods when working with zero-inflated data.

Furthermore, both ZIP and hurdle models provide more information than Poisson model when zero-inflation is present in the data. For example, income is associated with the odds of zero responses being collected from responders never went vs. have been on a boat trip but not when being sampled in ZIP. At the same time, income is associated with the odds of a person have not been to vs. have been to a boat trip. However, this information is not provided in Poisson models. In addition, even though the estimated βs are similar in all three models, the estimated β in Poisson model is larger than the values estimated in the ZIP and hurdle models.

This tutorial serves for the purpose of illustrating how the models can be estimated with `runjags`. Important topics such as model selections or other distributions handling count data are not covered in this paper. Readers can refer to Feng (2021) for a comprehensive comparison between ZIP and hurdle models handling zero-inflated count data.

References

Denwood, M. J. (2016). Runjags: An R package providing interface utilities, model templates, parallel computing methods and additional distributions for MCMC models in JAGS. *Journal of statistical software*, *71*, 1–25. doi: https://doi.org/10.18637/jss.v071.i09

Feng, C. X. (2021). A comparison of zero-inflated and hurdle models for modeling zero-inflated count data. *Journal of statistical distributions and applications*, *8*(1), 1–19. doi: https://doi.org/10.1186/s40488-021-00121-4

Gelman, A., & Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical science*, 457–472. doi: https://doi.org/10.1214/ss/1177011136

Grimm, K. J., & Stegmann, G. (2019). Modeling change trajectories with count and zero-inflated outcomes: Challenges and recommendations. *Addictive Behaviors*, *94*, 4–15. Retrieved 2022-09-04, from `https://linkinghub.elsevier.com/retrieve/pii/S0306460318310177` doi: https://doi.org/10.1016/j.addbeh.2018.09.016

Kleiber, C., & Zeileis, A. (2008). *Applied econometrics with R*. Springer-Verlag. Retrieved from `https://CRAN.R-project.org/package=AER`

Lambert, D. (1992). Zero-inflated Poisson regression, with an application to defects in manufacturing. *Technometrics : a journal of statistics for the physical, chemical, and engineering sciences*, *34*(1), 1–14. doi: https://doi.org/10.2307/1269547

Mullahy, J. (1986). Specification and testing of some modified count data models. *Journal of econometrics*, *33*(3), 341–365. doi: https://doi.org/10.1016/0304-4076(86)90002-3

Ntzoufras, I. (2011). *Bayesian modeling using WinBUGS*. John Wiley & Sons.

Plummer, M. (2003). JAGS: A program for analysis of Bayesian graphical models using Gibbs sampling. In *Proceedings of the 3rd international workshop on distributed statistical computing* (Vol. 124, pp. 1–10).

Team, R. C. (2013). *R: A language and environment for statistical computing.* Vienna, Austria.

Appendix A Supplemental Material

```
########### Get the data ready for analysis ###########

library(AER)
data("RecreationDemand")
dat<-RecreationDemand
dat$incomeC <- dat$income - mean(dat$income)
hist(dat$trips, main = "",
     xlab = "Number of recreational boating trips")

############### load packages ###############
library(runjags)
library(kableExtra)


############### analyses ###############

###### Poisson ######
Poisson_Model <- "model{
   ## Likelihood ##
   for (i in 1:N){
    Y[i] ~ dpois(lambda[i])
    log(lambda[i]) <- beta0 + beta1*X[i]
   }

   ## priors for coefficients
    beta0 ~ dnorm(0, 1/10000)
    beta1 ~ dnorm(0, 1/10000)

   ## exponentiate the paramters
   exp_beta0 <- exp(beta0)
   exp_beta1 <- exp(beta1)
}"

inits <- list(list(beta0 = rnorm(1, 0, 0.1),
              beta1 = rnorm(1, 0, 0.1)),
              list(beta0 = 1, beta1 = 1),
              list(beta0 = -1, beta1 = -1))

data <- list(Y = dat$trips, X = dat$incomeC,
             N = nrow(dat))

Pois_Est <- run.jags(model = Poisson_Model,
  monitor = c("beta0", "beta1", "exp_beta0",
  "exp_beta1"), data = data, n.chains = 3,
  inits = inits, method = "simple", adapt = 1000,
  burnin = 3000, sample = 10000)


res11<-cbind(round(Pois_Est$HPD[,c(1,3)],2),
  round(Pois_Est$summary$statistics[,1:2],2),
  round(Pois_Est$psrf$psrf[,1],4))
colnames(res11) <- c("Lower 95", "Upper 95",
  "Mean", "SD", "psrf")
kable(res11, caption = "Poisson runjags
  Output", "simple")
par(mfrow = c(1, 2))
plot(Pois_Est, plot.type = "trace")


###### ZIP ######
ZIP_model <- "model{
 ## likelihood
 for (i in 1:N){
  Y[i] ~ dpois(lambda[i])
  lambda[i] <- W[i]*mu[i]
  W[i] ~ dbern(1-p[i])
  log(mu[i]) <- beta0 + beta1*X[i]
  logit(p[i]) <- gamma0 + gamma1*B[i]
 }

 ## prior
  beta0 ~ dnorm(0, 1/10000)
  beta1 ~ dnorm(0, 1/10000)
  gamma0 ~ dnorm(0, 1/10000)
  gamma1 ~ dnorm(0, 1/10000)

  ## exponentiate the paramters
  exp_beta0 <- exp(beta0)
  exp_beta1 <- exp(beta1)
  exp_gamma0 <- exp(gamma0)
  exp_gamma1 <- exp(gamma1)
}"

W <- dat$trips
W[dat$trips>0] <- 1

inits <- list(list(beta0 = rnorm(1, 0, 0.1),
              beta1 = rnorm(1, 0, 0.1),
              gamma0 = rnorm(1, 0, 0.1),
              gamma1 = rnorm(1, 0, 0.1),
              W = W), list(beta0 = 1, beta1 = 1,
              gamma0 = 1, gamma1 = 1, W = W),
              list(beta0 = -1, beta1 = -1,
              gamma0 = 1,
              gamma1 = 1, W = W))

data <- list(Y = dat$trips, X = dat$incomeC,
             B = dat$incomeC, N = nrow(dat))

ZIP_Est <- run.jags(model = ZIP_model, monitor =
   c("beta0", "beta1", "gamma0", "gamma1",
   "exp_beta0", "exp_beta1", "exp_gamma0",
   "exp_gamma1"), data = data, n.chains = 3,
    inits = inits, method = "simple", adapt = 1000,
    burnin = 3000, sample = 10000,
    keep.jags.files = T, tempdir = T)
res2<-cbind(round(ZIP_Est$HPD[,c(1,3)],2),
  round(ZIP_Est$summary$statistics[,1:2],2),
  round(ZIP_Est$psrf$psrf[,1],4))
colnames(res2) <- c("Lower 95", "Upper 95", "Mean",
  "SD", "psrf")
res22<-round(exp(res2[,c(1, 2, 3)]),2)

kable(res22, caption = "ZIP runjags Exponentiated
  Output", "simple")
par(mfrow = c(1, 2))
plot(ZIP_Est, plot.type = "trace")


###### Hurdle ######
hurdle_model <- "model{
 ## likelihood
 C <- 10000
  for (i in 1:N){
  zeros[i] ~ dpois(-ll[i] + C)
  truncPois[i] <- Y[i]*log(mu[i]) - mu[i] -
        (log(1-exp(-mu[i])) + logfact(Y[i]))

  l1[i] <- (1-z[i])*log(p[i])
  l2[i] <- z[i]*(log(1-p[i]) + truncPois[i])
  ll[i] <- l1[i] + l2[i]

  log(mu[i]) <- beta0 + beta1*X[i]
  logit(p[i]) <- gamma0 + gamma1*B[i]
 }

 ## prior
  beta0 ~ dnorm(0, 1/10000)
  beta1 ~ dnorm(0, 1/10000)
  gamma0 ~ dnorm(0, 1/10000)
  gamma1 ~ dnorm(0, 1/10000)

 ## exponentiate the paramters
  exp_beta0 <- exp(beta0)
  exp_beta1 <- exp(beta1)
  exp_gamma0 <- exp(gamma0)
  exp_gamma1 <- exp(gamma1)
}"

z<-dat$trips
z[dat$trips > 0] <- 1

inits <- list(list(beta0 = rnorm(1, 0, 0.1),
                beta1 = rnorm(1, 0, 0.1),
                gamma0 = rnorm(1, 0, 0.1),
                gamma1 = rnorm(1, 0, 0.1)),
                list(beta0 = 1, beta1 = 1,
                gamma0 = 1, gamma1 = 1),
                list(beta0 = -1, beta1 = -1,
                gamma0 = 1,
                gamma1 = 1))

data <- list(Y = dat$trips, X = dat$incomeC,
                B = dat$incomeC,
                N = nrow(dat), z = z,
                zeros = rep(0, nrow(dat)))

hurdle_Est <- run.jags(model = hurdle_model,
                monitor = c("beta0", "beta1",
                 "gamma0", "gamm a1", "exp_beta0",
                "exp_beta1", "exp_gamma0", "exp_gamma1"),
                data = data, n.chains = 3,
                inits = inits, method = "simple",
                adapt = 1000,  burnin = 3000,
                sample = 10000,
                keep.jags.files = T, tempdir = T)

res33<-cbind(round(hurdle_Est$HPD[,c(1,3)],2),
          round(hurdle_Est$summary$statistics[,1:2],2),
          round(hurdle_Est$psrf$psrf[,1],4))
          colnames(res33) <- c("Mean", "SD",
          "Lower 95", "Upper 95","psrf")
res33<-round(exp(res33[,c(1, 2, 3)]),2)
kable(res33, caption = "Hurdle runjags
  Exponentiated Output", "simple")
plot(hurdle_Est, plot.type = "trace")
```