singleRcapture – an R package for single-source capture-recapture models

Piotr Chlebicki
Stockholm University
piotr.chlebicki@math.su.se

27.08.2024

Outline

Introduction

- This work is supported by the National Science Center, OPUS 20 grant no. 2020/39/B/HS4/00941 *Towards census-like statistics for foreign-born populations – quality, data integration and estimation*.
- The subject of this workshop is the `singleRcapture` package and its lightweight extension that allows for integration with other `R` packages called `singleRcaptureExtra`.
- The package is available on CRAN: `CRAN.R-project.org/package=singleRcapture` while the extension is available on: `https://github.com/ncn-foreigners/singleRcaptureExtra`.
- The GitHub page `github.com/ncn-foreigners/singleRcapture` always contains the current development version.

Estimating population size with only one register

Let Y_k represent the number of times k-th unit was observed in source data. Clearly, we don not know how often $Y_k = 0$ and to find the total population size N we need to estimate it. In general, we assume that conditional distribution of Y_k given a vector of covariates $\mathbf{x}_k$ follows some version of zero truncated count data distribution. Knowing the parameters of the distribution we may estimate the population size using Horwitz-Thompson type estimator:

$$\hat{N} = \sum_{k=1}^{N} \frac{\mathbb{1}(Y_k > 0)}{\mathbb{P}[Y_k > 0|\mathbf{x}_k]} = \sum_{k=1}^{N_{obs}} \frac{1}{\mathbb{P}[Y_k > 0|\mathbf{x}_k]},$$

and maximum likelihood estimate of N is obtained after substituting regression estimates for $\mathbb{P}[Y_k > 0|\mathbf{x}_k]$ into the equation above. Most of the methods relate to poisson processes.

The main function

The main function that `singleRcapture` is built around is `estimatePopsize`. The leading design principle was to make using `estimatePopsize` as close to standard `stats::glm` as possible. The most important parameters that must be supplied to `estimatePopsize` call are:

- `formula` – the main formula (i.e for the Poisson λ parameter),
- `data` – the `data.frame` (or `data.frame` coercible) object,
- `model` – either a function a string or a family class object specifying which model should be used possible values are listed in documentation. The supplied argument should have the form `model = "ztpoisson"`, `model = ztpoisson` or `model = ztpoisson(lambdaLink = "log")` the third way is the only one where the user may (but doesn't have to) select a link function.

The most important optional arguments include:

- `method` – numerical method used to fit regression `IRLS` or `optim`,
- `popVar` – a method for estimating variance of $\hat{N}$ and confidence interval creation (either bootstrap, analytic or skipping the estimation entirely),
- `controlMethod`, `controlModel`, `controlPopVar` – control parameters for numerical fitting, specifying additional formulas (inflation, dispersion) and population size estimation respectively. We will tackle these arguments separately,
- `offset` – a matrix of offset values with number of columns matching the number of distribution parameters providing offset values to each of linear predictors.

Supplying other arguments that work for `stats::glm` like `modelFrame`, `y`, `x`, `weights`, `subset`, `contrasts` is also possible.

Standard call for estimatePopsize

A package call used for fitting the zero-truncated poisson model from P. G. v. d. Heijden et al. 2003:

```
model <- estimatePopsize(
  formula = capture ~ gender + age + nation,
  data    = netherlandsimmigrant,
  popVar  = "analytic",
  model   = "ztpoisson",
  method  = "IRLS"
)
```

the netherlandsimmigrant data frame is exported by the package and contains data on unregistered immigration in four cities in Netherlands ($N_{obs} = 1880$), collected in 1995.

A one-inflated zero-truncated geometric model where the λ parameter depends only on age and the inflation parameter ω depends on the gender with the cloglog link, with bootstrap variance and custom instructions for model fitting for bootstrap samples:

```
modelInflated <- estimatePopsize(formula = capture ~ age,
    controlModel = controlModel(omegaFormula = ~ gender),
    data = netherlandsimmigrant, popVar = "bootstrap",
    model = oiztgeom(omegaLink = "cloglog"), method = "IRLS",
    controlPopVar = controlPopVar(B = 600, alpha = .01,
        bootType = "semiparametric",
        bootstrapFitcontrol = controlMethod(
          epsilon = 1e-6, silent = TRUE, stepsize = 2
        )
    )
)
```

Available models I

- Zero-truncated and zero-one-truncated Poisson, geometric, NB type II regression where the untruncated distribution is parameterized as:

$$\mathbb{P}[Y=y|\lambda,\alpha]=\frac{\Gamma\left(y+\alpha^{-1}\right)}{\Gamma\left(\alpha^{-1}\right)y!}\left(\frac{\alpha^{-1}}{\alpha^{-1}+\lambda}\right)^{\alpha^{-1}}\left(\frac{\lambda}{\lambda+\alpha^{-1}}\right)^{y}.$$

- Zero-truncated one-inflated (ztoi) modifications distributions where the new probability $\mathbb{P}^*$ measure is defined in terms of count data measure $\mathbb{P}$ with support on $\mathbb{N}\cup\{0\}$ as:

$$\mathbb{P}^*[Y=y]=\begin{cases}\mathbb{P}[Y=0] & y=0,\\ \omega\left(1-\mathbb{P}[Y=0]\right)+(1-\omega)\mathbb{P}[Y=1] & y=1,\\ (1-\omega)\mathbb{P}[Y=y] & y>1,\end{cases}$$

$$\mathbb{P}^*[Y=y|Y>0]=\omega\mathbb{1}_{\{1\}}(y)+(1-\omega)\mathbb{P}[Y=y|Y>0].$$

Available models II

- One-inflated zero-truncated (oizt) modifications distributions where the new probability $\mathbb{P}^*$ measure is defined as:

$$\mathbb{P}^*[Y=y] = \omega \mathbb{1}_{\{1\}}(y) + (1-\omega)\mathbb{P}[Y=y],$$

$$\mathbb{P}^*[Y=y|Y>0] = \omega \frac{\mathbb{1}_{\{1\}}(y)}{1-(1-\omega)\mathbb{P}[Y=0]} + (1-\omega)\frac{\mathbb{P}[Y=y]}{1-(1-\omega)\mathbb{P}[Y=0]}.$$

- Generalized Chao's and Zelterman's estimators via logistic regression on variable Z defined as $Z=1$ if $Y=2$ and $Z=0$ if $Y=1$ with $Z \sim b(p)$ where $\text{logit}(p) = \ln(\lambda/2)$ for poisson parameter λ,

$$\hat{N} = N_{obs} + \sum_{k=1}^{\boldsymbol{f}_1+\boldsymbol{f}_2} \left(2\exp\left(\boldsymbol{x}_k\hat{\boldsymbol{\beta}}\right) + 2\exp\left(2\boldsymbol{x}_k\hat{\boldsymbol{\beta}}\right)\right)^{-1}, \qquad \text{(Chao's estimator)}$$

$$\hat{N} = \sum_{k=1}^{N_{obs}} \left(1-\exp\left(-2\exp\left(\boldsymbol{x}_k\hat{\boldsymbol{\beta}}\right)\right)\right)^{-1}. \qquad \text{(Zelterman's estimator)}$$

Available models III

- Alternative approaches to modelling one-inflation that mimic hurdle models where the first type zero truncated hurdle model (ztHurdle) is defined as:

$$\mathbb{P}^{*}[Y=y]=\begin{cases}\frac{\mathbb{P}[Y=0]}{1-\mathbb{P}[Y=1]} & y=0,\\ \pi(1-\mathbb{P}[Y=1]) & y=1,\\ (1-\pi)\frac{\mathbb{P}[Y=y]}{1-\mathbb{P}[Y=1]} & y>1,\end{cases}$$

$$\mathbb{P}^{*}[Y=y|Y>0]=\pi\mathbb{1}_{\{1\}}(y)+(1-\pi)\mathbb{1}_{\mathbb{N}\setminus\{1\}}(y)\frac{\mathbb{P}[Y=y]}{1-\mathbb{P}[Y=0]-\mathbb{P}[Y=1]}$$

- The Hurdle zero truncarted (Hurdlezt) is defined as:

$$\mathbb{P}^{*}[Y=y]=\begin{cases}\pi & y=1,\\ (1-\pi)\frac{\mathbb{P}[Y=y]}{1-\mathbb{P}[Y=1]} & y\neq 1,\end{cases}\quad \mathbb{P}^{*}[Y=y|Y>0]=\begin{cases}\pi\frac{1-\mathbb{P}[Y=1]}{1-\mathbb{P}[Y=0]-\mathbb{P}[Y=1]} & y=1,\\ (1-\pi)\frac{\mathbb{P}[Y=y]}{1-\mathbb{P}[Y=0]-\mathbb{P}[Y=1]} & y>1.\end{cases}$$

Set-up

- As previously showcased the `singleRcapture` package supports modelling (linear) dependence on covariates of all parameters. To that end a modified IRLS algorithm is employed, full details are available in Yee 2015.
- In order to modify the `estimatePopsize` creates a modified model matrix $\boldsymbol{X}_{\text{vlm}}$
- In the context of multi-parameter families we have a matrix of linear predictors $\boldsymbol{\eta}$ instead of a vector, with the number of columns matching the number of parameters in the distribution.
- "Weights" in this modification are information matrices $\mathbb{E}\left[-\frac{\partial^2\ell}{\partial\boldsymbol{\eta}_{(k)}^T\partial\boldsymbol{\eta}_{(k)}}\right]$ where $\boldsymbol{\eta}_{(k)}$ is the k'th row of $\boldsymbol{\eta}$, while in the usual IRLS they are scalars $\mathbb{E}\left[-\frac{\partial^2\ell}{\partial\eta^2}\right]=-\frac{\partial^2\ell}{\partial\eta^2}$.

IRLS algorithm

1. Initialize with converged $\leftarrow$ FALSE, iter $\leftarrow 1, \boldsymbol{\eta} \leftarrow \texttt{start}, \boldsymbol{W} \leftarrow I, \ell \leftarrow \ell(\boldsymbol{\beta})$.
2. Store values from the previous step: $\ell_{-} \leftarrow \ell, \boldsymbol{W}_{-} \leftarrow \boldsymbol{W}, \boldsymbol{\beta}_{-} \leftarrow \boldsymbol{\beta}$ (the last assignment is omitted during the first iteration), and assign values in current iteration $\boldsymbol{\eta} \leftarrow \boldsymbol{X}_{\text{vlm}}\boldsymbol{\beta} + \boldsymbol{o}, \boldsymbol{W}_{(k)} \leftarrow \mathbb{E}\left[-\frac{\partial^2 \ell}{\partial \boldsymbol{\eta}_{(k)}^T \partial \boldsymbol{\eta}_{(k)}}\right], Z \leftarrow \boldsymbol{\eta}_{(k)} + \frac{\partial \ell}{\partial \boldsymbol{\eta}_{(k)}} \boldsymbol{W}_{(k)}^{-1} - \boldsymbol{o}_{(k)}$.
3. Assign current coefficient value: $\boldsymbol{\beta} \leftarrow (\boldsymbol{X}_{\text{vlm}} \boldsymbol{W} \boldsymbol{X}_{\text{vlm}})^{-1} \boldsymbol{X}_{\text{vlm}} \boldsymbol{W} \boldsymbol{Z}$.
4. If $\ell(\boldsymbol{\beta}) < \ell(\boldsymbol{\beta}_{-})$ try selecting the smallest value h such that for $\boldsymbol{\beta}_h \leftarrow 2^{-h}(\boldsymbol{\beta} + \boldsymbol{\beta}_{-})$ the inequality $\ell(\boldsymbol{\beta}_h) > \ell(\boldsymbol{\beta}_{-})$ holds if this is successful $\boldsymbol{\beta} \leftarrow \boldsymbol{\beta}_h$ else stop the algorithm.
5. If convergence is achieved end algorithm, else return to step 2.

Marginal frequencies

- A popular method of testing the model fit in single source capture-recapture studies is comparing the fitted marginal frequencies $\sum_{j=1}^{N_{obs}} \hat{\mathbb{P}}\left[Y_j = k | \boldsymbol{x}_j, Y_k > 0\right]$ with the observed marginal frequencies $\sum_{j=1}^{N} \mathbb{1}(Y_k = k) = \sum_{j=1}^{N_{obs}} \mathbb{1}(Y_k = k)$ for $k \geq 1$.
- If a fitted model bears sufficient resemblance to the real data collection process these quantities should be quite close.
- The `marginalFreq` function in `singleRcapture` is used for computing fitted marginal frequencies and the `summary` method can be called on a resulting object to perform G and χ^2 tests.

Diagnostic plots

The `singleRcapture` package allows for quickly making the following plots:

- `scaleLoc, hatplot, fitresid, qq` – These work exactly like their `stats::glm` counterparts, only for scale-location plot there are multiple linear predictors so there will be multiple plots.
- `marginal, rootogram` – Two ways of plotting fitted and observed marginal frequencies, for full exlanation of rootogram see Kleiber and Zeileis 2016.
- `bootHist` – Histogram of bootstrap statistics.
- `strata` – Plot of confidence intervals and point estimates for stratas.
- `dfpopContr, dfpopBox` – Plot of removal effect on the population size estimate against unit contribution (the inverse of inclusion probability).

All with just the standard `base` and `graphics` plotting, supplying additional arguments like `breaks` for histogram is possible.

Control parameters in `singleRcapture`

The `estimatePopsize` function accepts three different lists as control parameters allowing for great deal of customizability at call. These parameters are:

- `controlModel` – allows for specifying formulas on parameters other than the Poisson λ parameter and has an option of whether to use user supplied weights as counts,
- `controlMethod` –used for controlling the behaviour of the numeric algorithms used for regression fitting,
- `controlPopVar` – contains parameters for population size estimation and respective standard error and variance estimation.

Control parameters for method

The `controlMethod` argument admits the following parameters:

- `epsilon, maxiter, criterion` – tolerance for fitting algorithms by default 10^{-8}, maximum number of iterations (1000 for optim and 100 for IRLS by default) and convergence criterion,
- `coefStart, etaStart` – initial parameters, fitting via optim uses $\boldsymbol{\beta}$ while IRLS uses only $\boldsymbol{\eta}$,
- `optimMethod, optimPass` – method of `stats::optim()` used "Nelder-Mead" is the default and a list of control parameters to be passed into optim,
- `stepsize` – scaling of updates to $\boldsymbol{\beta}$ (by default 1),
- `verbose, printEveryN, saveIRLSlogs` – integer from 0 to 5 describing how much information should be printed, an integer controlling how often this information should be printed and a logical value indicating whether this infromation should be saved,

the best way of constructing a list with these parameters is to use `controlMethod` function. All of these are optional. Arguments not showcased here are in the `?controlMethod` section and are experimental or not important.

Control parameters for variance estimation

The `controlPopVar` argument admits the following parameters:

- `bootType`, `B`, `confType`, `keepbootStat` – type of bootstrap algorithm (parametric by default), number of bootstrap samples (500), type of confidence interval (percentile) and a logical value indicating whether bootstrap population size estimates should be saved in the object (`TRUE`),
- `fittingMethod`, `bootstrapFitcontrol` – fitting method (by default the same as used in the original call) and control parameters (controlMethod) for model fitting in bootstrap,
- `cores` – number of process cores to use in bootstrap (1 by default) parallel computing is done via `doParallel`, `foreach` and `parallel` packages,
- `traceBootstrapSize`, `bootstrapVisualTrace` – logical values indicating whether sample and population size should be tracked (`FALSE` by default) these work only when `cores` = 1,
- `alpha`, `sd`, `covType` – significance level (5% by default), type of standard deviation estimator to use and type of covariance matrix to use (either observed or fisher information matrix).

Semi-parametric bootstrap

The following bootstrap procedure assumes a accurate $\hat{N}$.

1. Draw the sample size $N'_{obs} \sim \text{Be}\left(N', \frac{N'-N_{obs}}{N'}\right)$, where $N' = \lfloor\hat{N}\rfloor + b\left(\lfloor\hat{N}\rfloor - \hat{N}\right)$.
2. Draw N'_{obs} units from the data uniformly without replacement.
3. Obtain new population size estimate using bootstrap data.
4. Repeat $1-3$ B times (by default 500).

Parametric bootstrap

The following bootstrap procedure assumes that supplied model is correct.

1. Draw the number of covariates equal to $\lfloor\hat{N}\rfloor + b\left(\lfloor\hat{N}\rfloor - \hat{N}\right)$ proportional to the estimated contribution $(\mathbb{P}\left[Y_k > 0|\boldsymbol{x}_k\right])^{-1}$ with replacement.
2. Using the fitted model and regression coefficients $\hat{\boldsymbol{\beta}}$ draw for each covariate the Y value from the corresponding probability measure on $\mathbb{N} \cup \{0\}$.
3. Truncate units with drawn Y value equal to 0.
4. Obtain population size estimate based on the truncated data.
5. Repeat $1 - 4$ B times.

Troubleshooting errors in `estimatePopsize`

When fitting in `estimatePopsize` fails for any reason the first action of the user should be to increase `verbose` parameter and rerun buggy code in order to gain more information about the problem. Then it is usually apparent what should be changed. Usually lowering the `stepsize` helps, if that fails consider trying different start values or look up `?controlMethod`.

Estimating population size with zerotrunc class objects

Using zerotrunc class objects from the countreg package in population size estimation is as easy as loading the singleRcaptureExtra extension and calling the estimatePopsize funcion on the desired object. For example fitting the previously mentioned model on immigration in Netherlands may be done with the following code:

```
model <- zerotrunc(
  formula = capture ~ gender + age + nation,
  data = netherlandsimmigrant,
  dist = "poisson"
)
estPop <- estimatePopsize(model)
```

Most of methods for objects created via estimatePopsize with formula will also work for estPop.

Additive models

Another package that is also integrated into `singleRcaptureExtra` is `VGAM`. This one is especially usefull if the assumption of linearity doesn't hold and approximating the regression function by polynomials is unlikely to result in a good model.
`VGAM` package also allows for linear models which are also integrated into `singleRcaptureExtra`.

Selected literature I

Heijden, Peter GM van der et al. (2003). “Point and interval estimation of the population size using the truncated Poisson regression model”. In: *Statistical Modelling* 3.4, pp. 305–322.

van der Heijden, Peter GM, Maarten Cruyff, and Hans C Van Houwelingen (2003). “Estimating the size of a criminal population from police records using the truncated Poisson regression model”. In: *Statistica Neerlandica* 57.3, pp. 289–304.

Cruyff, Maarten J. L. F. and Peter G. M. van der Heijden (2008). “Point and Interval Estimation of the Population Size Using a Zero-Truncated Negative Binomial Regression Model”. In: *Biometrical Journal* 50.6, pp. 1035–1050.

Selected literature II

Böhning, Dankmar and Peter G. M. van der Heijden (2009). “A covariate adjustment for zero-truncated approaches to estimating the size of hidden and elusive populations”. In: *The Annals of Applied Statistics* 3.2, pp. 595–610. DOI: 10.1214/08-AOAS214.

Böhning, Dankmar, Alberto Vidal-Diez, et al. (2013). “A Generalization of Chao’s Estimator for Covariate Information”. In: *Biometrics* 69.4, pp. 1033–1042.

Yee, Thomas W. (2015). *Vector Generalized Linear and Additive Models: With an Implementation in R*. 1st. Springer Publishing Company, Incorporated.

Selected literature III

Kleiber, Christian and Achim Zeileis (2016). "Visualizing Count Data Regressions Using Rootograms". In: *The American Statistician* 70.3, pp. 296–303. DOI: 10.1080/00031305.2016.1173590.

Godwin, Ryan T. and Dankmar Böhning (2017). "Estimation of the population size by using the one-inflated positive Poisson model". In: *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 66.2, pp. 425–448.

Selected literature IV

Böhning, Dankmar and Peter G. M. van der Heijden (2019). “The identity of the zero-truncated, one-inflated likelihood and the zero-one-truncated likelihood for general count densities with an application to drink-driving in Britain”. In: *The Annals of Applied Statistics* 13.2, pp. 1198–1211. DOI: 10.1214/18-AOAS1232.