Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. doi: 10.18637/jss.v000.i00

Single-Source Capture-Recapture Models With singleRcapture

Piotr Chlebicki
Stockholm University

Maciej Beręsewicz
Poznań University of Economics and Business

Abstract

Estimating population size is an important issue in official statistics, social sciences and natural sciences. One way to approach this problem is to use capture-recapture methods, which can be classified according to the number of sources used, the main distinction being between methods based on one source and those based on two or more sources. In this presentation we will introduce the **singleRcapture** R package for fitting SSCR models. The package implements state-of-the-art models as well as some new models proposed by the authors (e.g. extensions of zero-truncated one-inflated and one-inflated zero-truncated models). The software is intended for users interested in estimating the size of populations, particularly those that are difficult to reach or for which information is available from only one source and dual/multiple system estimation cannot be used.

Keywords: population size estimation, truncated distributuons, count regression models, R.

1. Introduction

1.1. Literature review

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 **singleRcapture** package is an R language package that focuses on implementing state of the art methods for frequentist point and interval estimation of size of closed populations in single-source capture-recapture (SSCR) setting (e.g. estimation of the population size of irregular migrants at set time point in a given area).

The beginning of inference in single source capture-recapture dates back to the seminal van der Heijden, Bustami, Cruyff, Engbersen, and van Houwelingen (2003) paper in which the zero truncated poisson model was applied to study the size of population of irregular migrants in fours cities in Netherlands.

1.2. How do we estimate 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:

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

where $I_k := \mathcal{I}_{\mathbb{N}}(Y_k)$, 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 analytic variance estimation is then done by computing two parts of the decomposition due to the law of total variance:

$$\text{var}[\hat{N}] = \mathbb{E}\left[\text{var}\left[\hat{N}|I_1, \ldots, I_n\right]\right] + \text{var}\left[\mathbb{E}[\hat{N}|I_1, \ldots, I_n]\right]$$

where the first addend is by the multivariate δ method seen to be:

$$\mathbb{E}\left[\text{var}\left[\hat{N}|I_1, \ldots, I_n\right]\right] = \left(\frac{\partial(N|I_1, \ldots, I_N)}{\partial\boldsymbol{\beta}}\right)^T \text{cov}\left[\boldsymbol{\beta}\right] \left(\frac{\partial(N|I_1, \ldots, I_N)}{\partial\boldsymbol{\beta}}\right)\Bigg|_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}}$$

while the later part of the decomposition is under the assumption of independence of I_k's optimally estimated via:

$$\begin{aligned}
\text{var}\left(\mathbb{E}(\hat{N}|I_1, \ldots, I_n)\right) &= \text{var}\left(\sum_{k=1}^{N} \frac{I_k}{\mathbb{P}(Y_k > 0)}\right) \\
&= \sum_{k=1}^{N} \text{var}\left(\frac{I_k}{\mathbb{P}(Y_k > 0)}\right) \\
&= \sum_{k=1}^{N} \frac{1}{\mathbb{P}(Y_k > 0)^2} \text{var}(I_k) \\
&= \sum_{k=1}^{N} \frac{1}{\mathbb{P}(Y_k > 0)^2} \mathbb{P}(Y_k > 0)(1 - \mathbb{P}(Y_k > 0)) \\
&= \sum_{k=1}^{N} \frac{1}{\mathbb{P}(Y_k > 0)} (1 - \mathbb{P}(Y_k > 0))
\end{aligned}$$

$$
\begin{aligned}
&\approx \sum_{k=1}^{N} \frac{I_k}{\mathbb{P}(Y_k > 0)^2}(1 - \mathbb{P}(Y_k > 0)) \\
&= \sum_{k=1}^{N_{obs}} \frac{1 - \mathbb{P}(Y_k > 0)}{\mathbb{P}(Y_k > 0)^2}
\end{aligned}
\tag{1}
$$

1.3. Example with R code

Installation:

```
R> install.packages("singleRcapture")
R> remotes::install_github(
+   "https://github.com/ncn-foreigners/singleRcaptureExtra"
+ )

R> library(singleRcapture)
R>
R> head(netherlandsimmigrant)

  capture gender    age       reason       nation
1       1   male <40yrs Other reason North Africa
2       1   male <40yrs Other reason North Africa
3       1   male <40yrs Other reason North Africa
4       1   male <40yrs Other reason         Asia
5       1   male <40yrs Other reason         Asia
6       2   male <40yrs Other reason North Africa

R> summary(netherlandsimmigrant)

    capture         gender         age                reason
 Min.   :1.000   female: 398   <40yrs:1769   Illegal stay: 259
 1st Qu.:1.000   male  :1482   >40yrs: 111   Other reason:1621
 Median :1.000
 Mean   :1.162
 3rd Qu.:1.000
 Max.   :6.000
                   nation
 American and Australia: 173
 Asia                  : 284
 North Africa          :1023
 Rest of Africa        : 243
 Surinam               :  64
 Turkey                :  93

R> basicModel <- estimatePopsize(
+   formula = capture ~ gender + age + nation,
+   model   = ztpoisson(),
+   data    = netherlandsimmigrant
+ )
R>
R> summary(basicModel)


Call:
estimatePopsize.default(formula = capture ~ gender + age + nation,
    data = netherlandsimmigrant, model = ztpoisson())

Pearson Residuals:
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
-0.486442 -0.486442 -0.298080  0.002093 -0.209444 13.910844

Coefficients:
-----------------------
For linear predictors associated with: lambda
                     Estimate Std. Error z value  P(>|z|)
(Intercept)           -1.3411     0.2149  -6.241 4.35e-10 ***
gendermale             0.3972     0.1630   2.436 0.014832 *
age>40yrs             -0.9746     0.4082  -2.387 0.016972 *
nationAsia            -1.0926     0.3016  -3.622 0.000292 ***
nationNorth Africa     0.1900     0.1940   0.979 0.327398
nationRest of Africa  -0.9106     0.3008  -3.027 0.002468 **
nationSurinam         -2.3364     1.0136  -2.305 0.021159 *
nationTurkey          -1.6754     0.6028  -2.779 0.005445 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

AIC: 1712.901
BIC: 1757.213
Residual deviance: 1128.553

Log-likelihood: -848.4504 on 1872 Degrees of freedom
Number of iterations: 8
-----------------------
Population size estimation results:
Point estimate 12690.35
Observed proportion: 14.8% (N obs = 1880)
Std. Error 2808.169
95% CI for the population size:
          lowerBound upperBound
normal      7186.444   18194.26
logNormal   8431.275   19718.32
95% CI for the share of observed population:
          lowerBound upperBound
```

```
normal      10.332927   26.16037
logNormal    9.534281   22.29793
```

2. Detailed information

2.1. Fitting method

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 employ the algorithm a modified model matrix is created $\boldsymbol{X}_{\text{vlm}}$ at call to `estimatePopsize`. In the context of the models implemented in **singleRcapture** this matrix can be writen as:

$$\boldsymbol{X}_{vlm} = \begin{pmatrix} \boldsymbol{X}_1 & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{X}_2 & \dots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \dots & \boldsymbol{X}_p \end{pmatrix} \tag{2}$$

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" are then modified to be 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 $\mathbb{E}\left[-\frac{\partial^2\ell}{\partial\eta_k^2}\right]$ which is often just $-\frac{\partial^2\ell}{\partial\eta^2}$.

1. Initialize with `converged` $\leftarrow$ FALSE, `iter` $\leftarrow 1, \boldsymbol{\eta} \leftarrow$ `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: $\boldsymbol{\beta} \leftarrow \left(\boldsymbol{X}_{\text{vlm}}\boldsymbol{W}\boldsymbol{X}_{\text{vlm}}\right)^{-1}\boldsymbol{X}_{\text{vlm}}\boldsymbol{W}\boldsymbol{Z}$.
4. If $\ell(\boldsymbol{\beta}) < \ell(\boldsymbol{\beta}_{-})$ try selecting the smallest value h such that for $\boldsymbol{\beta}_h \leftarrow 2^{-h}\left(\boldsymbol{\beta} + \boldsymbol{\beta}_{-}\right)$ 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.

2.2. Avaiable models

The full list of implemented models in **singleRcapture** along with the expressions for probability density functions and point estimates is found in the collective help file for all family functions:

```
R> ?ztpoisson
```

Here we limit ourselves to just listing the family functions:

- 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:

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

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

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

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

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

- 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,

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

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

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

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

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

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

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

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

2.3. Structure of a family function

- `makeMinusLogLike` – A factory function for creating the:

  $$\ell(\boldsymbol{\beta}), \frac{\partial \ell}{\partial \boldsymbol{\beta}}, \frac{\partial^2 \ell}{\partial \boldsymbol{\beta}^T \partial \boldsymbol{\beta}}$$

  functions from $\boldsymbol{y}$ vector and $\boldsymbol{X}_{vlm}$ the argument `deriv` with possible values in `c(0, 1, 2)` provides which derivative to return with the default `0` being just the minus log-likelihood.

- `links` – List with link functions.

- `mu.eta, variance` – Functions of linear predictors that return expected value and variance. There is a ‘type‘ argument with 2 possible values `"trunc"` and `"nontrunc"` that specifies whether to return $\mathbb{E}[Y|Y>0], \mathrm{var}[Y|Y>0]$ or $\mathbb{E}[Y], \mathrm{var}[Y]$ respectively, also the `deriv` argument with values in `c(0, 1, 2)` is used for indicating the derivative with respect to the linear predictors with is used for providing standard error in `predict` method.

- `family` – Character that specifies name of the model.

- `valideta, validmu` – For now only returns true. In near future will be used to check whether applied linear predictors are valid (i.e. are transformed into some elements of parameter space the subjected to inverse link function).

- `funcZ, Wfun` – Functions that create pseudo residuals and working weights used in IRLS algorithm.

- `devResids` – Function that given the linear predictors prior weights vector and response vector returns deviance residuals.

- `pointEst, popVar` – Functions that given prior weights linear predictors and in the later case also estimation of $\mathrm{cov}(\hat{\boldsymbol{\beta}})$ and $\boldsymbol{X_{vlm}}$ matrix return point estimate for population size and analytic estimation of its variance.There is a additional boolean parameter `contr` in the former function that if set to true returns contribution of each unit.

- `etaNames` – Names of linear predictors.

- `densityFunction` – A function that given linear predictors returns value of PMF at values `x`. Additional argument `type` specifies whether to return $\mathbb{P}[Y|Y>0]$ or $\mathbb{P}[Y]$.

- `simulate` – A function that generates values of dependent vector given linear predictors.
- `getStart` – Expression for generating starting points.

2.4. Marginal frequencies

A popular method of testing the model fit in single source capture-recapture studies is comparing the fitted marginal frequencies $\sum_{j=1}^{N_{obs}} \hat{\mathbb{P}}\left[Y_j = k|\boldsymbol{x}_j, Y_j > 0\right]$ with the observed marginal frequencies $\sum_{j=1}^{N} \mathcal{I}_{\{k\}}(Y_k) = \sum_{j=1}^{N_{obs}} \mathcal{I}_{\{k\}}(Y_k)$ for $k \geq 1$. If a fitted model bears sufficient resemblance to the real data collection process these quantities should be quite close and both G and χ^2 tests may be employed in order to test the statistical significance of the discrepancy with the following **singleRcapture** syntax:

```
R> (margFreq <- marginalFreq(basicModel))

$table
           0            1            2            3            4            5
1.081035e+04 1.612589e+03 2.337193e+02 3.013473e+01 3.242543e+00 2.908758e-01
           6
2.214773e-02

$y
y
   1    2    3    4    5    6
1645  183   37   13    1    1

$df
[1] -3

$name
[1] "ztpoisson"

attr(,"class")
[1] "singleRmargin"

R> summary(margFreq, df = 1, dropl5 = "group")

Test for Goodness of fit of a regression model:

                 Test statistics df P(>X^2)
Chi-squared test           50.06  1 1.5e-12
G-test                     34.31  1 4.7e-09

--------------------------------------------------------------
```

```
Cells with fitted frequencies of < 5 have been grouped
Names of cells used in calculating test(s) statistic: 1 2 3
```

where the `dropl5` argument is used to indicate how to handle the cells with less than 5 fitted observations, note however that currently there is no continuity correction.

References

van der Heijden PG, Bustami R, Cruyff MJ, Engbersen G, van Houwelingen HC (2003). "Point and interval estimation of the population size using the truncated Poisson regression model." *Statistical Modelling*, **3**(4), 305–322.

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

Affiliation:

Piotr Chlebicki
Stockholm University
Matematiska institutionen
106 91 Stockholm, Albano hus 1
E-mail: piotr.chlebicki@math.su.se
URL: https://github.com/Kertoo

Maciej Beręsewicz
Poznań University of Economics
and Business
Poznań University of Economics and Business
Department of Statistics
Institute of Informatics and Quantitative Economics
Al. Niepodległosci 10
61-875 Poznań, Poland
E-mail: maciej.beresewicz@ue.poznan.pl

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
MMMMMM YYYY, Volume VV, Issue II *Submitted:* yyyy-mm-dd
doi:10.18637/jss.v000.i00 *Accepted:* yyyy-mm-dd