A simulation study exploring weighted likelihood models to recover unbiased population estimates from weighted survey data.

WorldPop, University of Southampton

06 February 2021

Contents

1 Introduction

> Note: This report assumes familiarity with Bayesian statistical models and notation, Stan and JAGS software, and the R statistical programming language.

Statistical models used to map population estimates across the landscape require observations of population counts from a representative sample of locations to use as training data. These data usually come from household surveys in which populations are enumerated within geographically-defined survey locations. A stratified random sample is ideal for recovering unbiased estimates of the mean and variance of population densities. However, household surveys often implement a PPS sampling design (Probability Proportional to Size) in which locations with higher population densities are more likely to be included in the sample. This will result in biased estimates of average population densities for population modelling. Population-weighted sampling is intended to approximate random samples of *individuals* or *households* from sets of geographically clustered households, but it does not produce random samples of *locations* needed for geographical population models.

Our objectives here are to:

1. Demonstrate that a population-weighted sample results in biased estimates of population densities,
2. Demonstrate that model-based estimates of population totals for large areas are sensitive to this bias,
3. Explore Bayesian weighted-likelihood and weighted-precision approaches to produce unbiased parameter estimates, and
4. Demonstrate that weighted models can recover unbiased estimates of population densities and population totals from a population-weighted sample.

This analysis is intended as a theoretical foundation to support ongoing development of statistical models to estimate and map population sizes using weighted survey data as inputs.

2 Methods

We simulated populations by drawing population densities for each location from a distribution with known parameters. We then produced various types of samples from those populations: random, population-weighted, or a combination. Every population included one million locations and every sample included 2000 locations. We fit three types of models to these data trying to recover the known population parameters: unweighted model, weighted-precision model, and weigthed-likelihood model.

All simulations were conducted using the R statistical programming environment (R Core Team 2020). Statistical models were fit using either the *RStan* R package (Stan Development Team 2019, 2020) or the *runjags* R package (Plummer & others 2003, Denwood 2016).

2.1 Simulated Populations

We used a log-normal distribution to represent population densities following the population model of Leasure et al (2020):

$$\begin{aligned} N_i &\sim Poisson(D_i A_i) \\ D_i &\sim LogNormal(\mu_i, \sigma_{t,g}) \\ \mu_i &= \alpha_{t,g} + \sum_{k=1}^{K} \beta_k x_{k,i} \end{aligned} \tag{1}$$

In this model, N_i was the observed population count and A_i was the observed settled area (ha) at location i. Population densities D_i were modelled as a function of settlement types t (e.g. urban/rural), geographic units g, and K geospatial covariates $x_{k,i}$. The regression parameters $\alpha_{t,g}$, β_k, and $\sigma_{t,g}$ estimated average population densities, effects of covariates, and unexplained residual variation, respectively.

The intended purpose of this model was to estimate model parameters based on observed population data. For the purposes of the current simulation study, we reversed that logic; We will provide pre-defined parameter values to generate simulated population data.

For our simulations we made a series of simplifying assumptions to this model. We assumed that every location i included one hectare of settled area (i.e. $A_i = 1$) and we ignored the Poisson variation so that $N_i = D_i$. We also ignored the effects of settlement type, geographic location, and covariates so that t, g, and $x_{k,i}$ were dropped from the model. These simplifying assumptions allowed us to isolate the effects of weighted sampling in the absence of these potentially confounding effects. While beyond the scope of the current report, relaxing these assumptions and assessing their effects should be the focus of future theoretical and empirical studies.

The simplified model used for our simulations was:

$$D_i \sim LogNormal(log(\mu), \sigma) \tag{2}$$

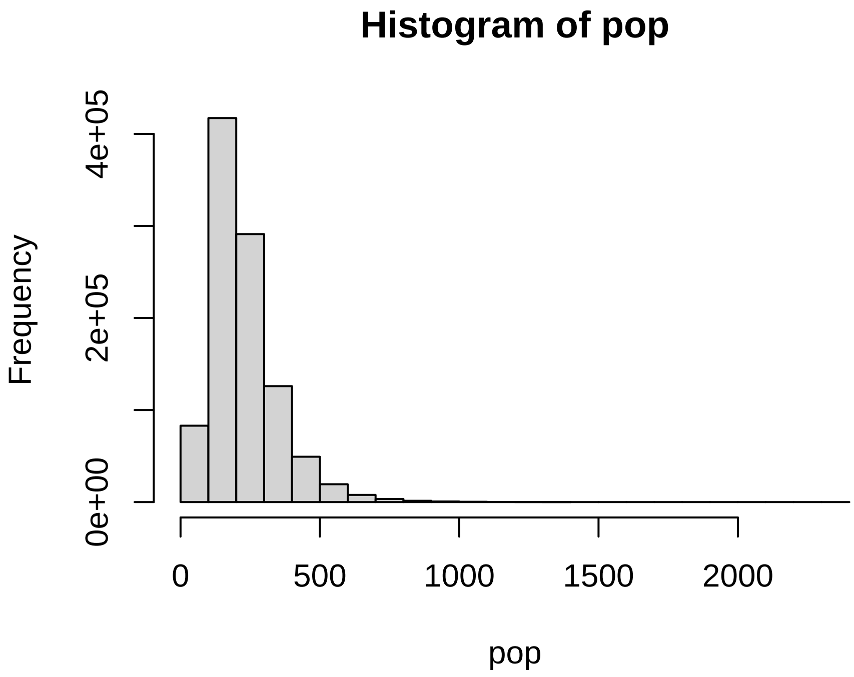
> Note: We modelled the median μ on the natural scale so that the parameter estimates were easier to interpret, but we kept σ on the log-scale to simplify the equations.

We simulated population densities (i.e. people per hectare) at one million locations by taking one million draws from this log-normal distribution using a range of parameter values for μ (i.e. 100, 250, 500) and σ (i.e. 0.25, 0.5, 0.75).

Following Eq. (2), a population where $\mu = 250$ and $\sigma = 0.5$ can be simulated across one million locations using the following R code:

```
# simulate population densities at one million locations
pop <- rlnorm(n = 1e6,
              meanlog = log(200),
              sdlog = 0.5)

# plot distribution of population densities
hist(pop)
```



2.2 Simulated Survey Data

We simulated three sampling designs:

1. Random sampling,

2. Population-weighted sampling, and
3. A combination of random and population-weighted sampling.

We always used a sample size of 2000 locations. We used the following R code to draw these samples.

2.2.1 Random Sample

The random sample was simply drawn using the *sample* function to draw 2000 samples without replacement from the simulated population densities:

```r
# random sample
D <- sample(x = pop,
            n = 2e3)
```

2.2.2 Population-weighted Sample

To draw a population-weighted sample, we must first calculate sampling probabilities that are based on the population density at each location.

```r
# sampling weights based on population density
w <- pop / sum(pop)

# select locations for a weighted sample
i <- sample(x = 1:length(pop),
            size = 2e3,
            prob = w)

# population densities at selected locations
D <- pop[i]
```

2.2.3 Combined Sample

Combined samples (random and weighted) were produced using several different proportions of random samples (i.e. 0.2, 0.5, 0.8). The total sample size for a combined sample was still 2000.

```r
# proportion random
prop <- 0.5

# select locations for weighted sample
i <- sample(x = 1:length(pop),
            size = 2e3*(1-prop),
            prob = w)

# select locations for random sample
j <- sample(x = 1:length(pop)[-i],
            size = 2e3*prop)

# weights for selected locations in weighted sample
w_i <- w[i]

# weights for selected locations in random sample
w_j <- rep(x = mean(w_i),
           times = n*prop)
```

```r
# population densities at selected locations
D <- pop[ c(i,j) ]
```

Notice that we assigned equal weights to all of the random samples that were equal to the mean weight among the weighted samples. In other words, each random sample was given an equal weight in the model comparable to an average weighted sample to balance the influence of these portions of the sample.

2.3 Statistical Models

We evaluated four statistical models:

1. Unweighted log-normal model (Stan),
2. Weighted-likelihood log-normal model (Stan), and
3. Weighted-precision log-normal model (Stan and JAGS).

The unweighted model was included to evaluate the bias that arises when fitting an unweighted model to weighted sample data. The weighted-precision and weighted-likelihood models were designed to use sample weights to recover unbiased parameter estimates from a weigthed sample. We developed the weighted-precision model for both Stan and JAGs to demonstrate that both implementations produced the same results and to provide example code for both software packages. The weighted-likelihood approach required a direct adjustment to the likelihood that was not possible to implement in JAGS.

2.3.1 Unweighted Log-normal

Our simplest model was a log-normal with no weights:

$$D_i \sim LogNormal(log(\mu), \sigma) \tag{3}$$

Notice that this is identical to Eq. (2) that was used to generate our simulated populations. Our implementation used the following Stan model:

```
data{
  int<lower=0> n;         // sample size
  vector<lower=0>[n] D;   // observed population densities
}

parameters{
  real<lower=0> mu;      // median (natural)
  real<lower=0> sigma;   // standard deviation (log)
}

model{
  D ~ lognormal(log(mu), sigma);  // likelihood

  mu ~ uniform(0, 2e3);   // prior for mu
  sigma ~ uniform(0, 5);  // prior for sigma
}
```

2.3.2 Weighted-likelihood

The weighted-likelihood approach used the same log-normal model but implemented an adjustment to the likelihood for each sample based on the sample weights to account for the increased probability of including locations with high population densities in the weighted sample. We implemented this model in Stan:

```
data{
  int<lower=0> n;                  // sample size
  vector<lower=0>[n] D;            // observed population densities
  vector<lower=0,upper=1>[n] w;  // sampling probabilities (weights)
}

parameters{
  real<lower=0> mu;      // median (natural)
  real<lower=0> sigma;   // standard deviation (log)
}

model{

  // weighted likelihood
  for(i in 1:n){
    target += lognormal_lpdf( D[i] | log(mu), sigma ) / w[i];
  }

  mu ~ uniform(0, 2e3);   // prior for mu
  sigma ~ uniform(0, 5);  // prior for sigma
}
```

> Note: The sampling probabilities `w` were defined in the section above (see Simulated Survey Data).

In this model, the likelihood for each sample is divided by its sampling probability–the probability of a location being selected for the sample out of the one million locations in the population. This adjustment to the likelihood reduces the influence on parameter estimates of locations that had higher sampling probabilities (i.e. locations with high population densities are over-represented in a population-weighted sample). If this model were used for a random sample, all of the weights would be equal and it would be equivalent to the unweighted model above (see Unweighted Log-normal).

2.3.3 Weighted-precision (JAGS)

A potential alternative to the weighted-likelihood approach would be to scale the precision τ of the log-normal using the location-specific weights w_i. Precision τ is defined as the inverse of variance σ^2:

$$\begin{gathered} \tau = \sigma^{-2} \\ \sigma = \sqrt{\tau^{-1}} \end{gathered} \tag{4}$$

For this model, we need to define an inverse sampling weight m_i that is scaled to sum to one across all samples:

$$m_i = \frac{w_i^{-1}}{\sum_{i=1}^{n} w_i^{-1}} \tag{5}$$

We will refer to these scaled invserse sampling weights as model weights m_i. Now we can specify a weighted-precision model as:

$$\begin{aligned} D_i &\sim LogNormal(\mu_i, \sqrt{\tau_i^{-1}}) \\ \tau_i &= \theta^{-2} m_i \end{aligned} \tag{6}$$

where θ^{-2} is a naive precision that does account for the model weights m_i. Notice that the precision τ_i is location-specific (i.e. indexed by i) after it has been adjusted by the model weights m_i, and that $\sqrt{\tau_i^{-1}}$ is a location-specific weighted standard deviation. Where the model weights are relatively low, the location-specific precisions τ_i will be decreased to reduce the weight of those samples in the model. For our population-weighted sample, this will reduce the weight of locations with high population densities that were over-represented in the sample.

Our goal is to recover an unbiased estimate of the standard deviation of population densities among all locations in the population. So far, we have only produced location-specific estimates of precision τ_i that are dependent on location-specific model weights. We derived the global standard deviation σ using a weighted average of the location-specific standard deviations $\sqrt{\tau_i^{-1}}$:

$$\sigma = \frac{\sum_{i=1}^{n} \sqrt{\tau_i^{-1}} \sqrt{m_i}}{\sum_{i=1}^{n} \sqrt{m_i}} \tag{7}$$

We will switch to JAGS for this model because it parameterizes the log-normal distribution using precision rather than standard deviation (as in Stan) and this makes the implementation of a weighted-precision model simpler. In the next section, we will also implement the weighted-precision model in Stan.

```
model{

  for(i in 1:n){

    # likelihood
    D[i] ~ dlnorm(log(mu), tau[i])

    # location-specific weighted precision
    tau[i] <- pow(theta,-2) * m[i]
  }

  # prior for median
  mu ~ dunif(0, 2e3)

  # prior for naive standard deviation
  theta ~ dunif(0, 1)

  # scaled inverse sampling weight
  m <- pow(w,-1) / sum(pow(w,-1))

  # weighted average sigma
  sigma <- sum( sqrt( 1 / tau ) * sqrt(m) ) / sum( sqrt(m) )
}
```

2.3.4 Weighted-precision (Stan)

We also implemented the weighted-precision model in Stan:

```
data{
  int<lower=0> n;                // sample size
  vector<lower=0>[n] D;          // observed counts
  vector<lower=0,upper=1>[n] w;  // sampling probabilities
}

transformed data{

  // scaled inverse weights
  vector<lower=0,upper=1>[n] m = inv(w) ./ sum(inv(w));
}

parameters{
  real<lower=0> mu;     // median
  real<lower=0> theta;   // naive standard deviation
}

transformed parameters{

  // location-specific weighted sigma
  vector<lower=0>[n] sigma_i = sqrt( inv( m * pow(theta,-2) ) );
}

model{

  // likelihood
  D ~ lognormal( log(mu), sigma_i );

  // priors
  mu ~ uniform(0, 2e3);
  theta ~ uniform(0, 1);
}

generated quantities {

  // weighted average sigma
  real<lower=0> sigma = sum( sigma_i .* sqrt(m) ) / sum( sqrt(m));
}
```

3 Results

This section should describe the results of the analysis.

Links to the full results should be provided when possible (e.g. WOPR, woprVision, DOI link, etc).

Sufficient diagnostics should be included to demonstrate that the results are fit for the intended uses.

4 Discussion

This section should describe the known limitations and assumptions associated with the results.

This section should recommend next steps that the authors see as most promising to build on the work.

Contributing

This section should describe funding sources, authors, contributers, external collaborators, etc. and their contributions.

Suggested Citation

Leasure DR, Dooley CA, Tatem AJ. 2021. A simulation study exploring weighted likelihood models to recover unbiased population estimates from weighted survey data. WorldPop, University of Southampton. doi:10.5258/SOTON/WP00XXX

License

References

Denwood MJ. 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:10.18637/jss.v071.i09.

Leasure DR, Jochem WC, Weber EM, Seaman V, Tatem AJ. 2020. National population mapping from sparse survey data: A hierarchical bayesian modeling framework to account for uncertainty. *Proceedings of the National Academy of Sciences.*

Plummer M, others. 2003. JAGS: A program for analysis of bayesian graphical models using gibbs sampling. In: *Proceedings of the 3rd international workshop on distributed statistical computing.* Vienna, Austria., 1–10. http://mcmc-jags.sourceforge.net/.

R Core Team. 2020. *R: A language and environment for statistical computing.* Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Stan Development Team. 2019. *Stan user's guide.* https://mc-stan.org/docs/2_23/stan-users-guide.

Stan Development Team. 2020. *RStan: The r interface to stan. r package version 2.19.3.* http://mc-stan.org/.

Appendix A: Supplementary Files

Model Data

This sub-section describes all input data published as supplementary information.

Model Code

This sub-section describes all code published as supplementary information.