Using R for Bayesian Analyses of Survey Data

Matt Williams\textsuperscript{1} Terrance Savitsky\textsuperscript{2}

\textsuperscript{1}National Center for Science and Engineering Statistics
National Science Foundation
mrwillia@nsf.gov

\textsuperscript{2}Office of Survey Methods Research
Bureau of Labor Statistics
Savitsky.Terrance@bls.gov

Government Advances in Statistical Programming
October 25, 2018
Overview

- Bayesian inference from survey samples
- Fitting Bayesian models with Stan
- Plotting results with ggplot2
Inference from Survey Samples

- **Goal of Analyst**: perform inference about a finite population generated from an unknown model, \( P_0 \).

- **Data Collected**: from under a complex sampling design distribution, \( P_\nu \)
  
  - Probabilities of inclusion \( \pi_i \) are often associated with the variable of interest (purposefully)
  
  - Sampling designs are “informative”: the balance of information in the sample \( \neq \) balance in the population.

- **Biased Estimation**: estimate \( P_0 \) without accounting for \( P_\nu \).
  
  - Use inverse probability weights \( w_i = 1/\pi_i \) to mitigate bias.
The Pseudo-Posterior Estimator

The plug-in estimator for posterior density under the analyst-specified model for $\lambda \in \Lambda$ is

$$\hat{\pi} (\lambda | y_o, w) \propto \left[ \prod_{i=1}^{n} p(y_o, i | \lambda)^{w_i} \right] \pi (\lambda),$$

- **pseudo-likelihood**: $\prod_{i=1}^{n} p(y_o, i | \lambda)^{w_i}$
- **prior**: $\pi (\lambda)$
- **values** $y_o$ and **sampling weights** $\{w\}$ for individuals observed in sample

We are going to use **Stan** to estimate $\hat{\pi} (\lambda | y_o, w)$. 
Related Papers

- **Consistency** of the Pseudo-Posterior
  - Savitsky and Toth (2016)
- Extension to **Divide and Conquer** methods
  - Savitsky and Srivastava (2018)
- **Joint** modelling of Outcome and Weights
  - Novelo and Savitsky (2017)
- Extension to **pairwise weights and outcomes**
  - Williams and Savitsky (2018b)
- Extension to **multistage surveys**
  - Williams and Savitsky (2018a)
- **Correction of asymptotic coverage**
  - Williams and Savitsky (2018c)
Stan

- Stan is a platform for **statistical modeling and computation** (Stan Development Team, 2016)
  - Users specify log density functions
  - Stan provides **MCMC sampling**, variational inference, or maximum likelihood optimization
  - Stan interfaces with several languages, including R (**Rstan**)
    - Requires **Rtools**, for compiling of C++ code.
- We use Stan for
  - survey weighted **logistic regression**
  - survey weighted **quantile regression with penalized splines**
Stan: Files

R file (.R)

library(rstan)
# compile stan code
mod = stan_model('wt_logistic.stan')
#sample stan model, given data, other inputs
sampling(object = mod, data = ...)

Stan file (.stan)

functions{ }
data{ }
parameters{ }
transformed parameters{ }
model{ }
functions{
    real wt_bin_lpmf(int[] y, vector mu, vector weights, int n){
        real check_term;
        check_term = 0.0;
        for( i in 1:n )
        {
            check_term = check_term +
            weights[i] * bernoulli_logit_lpmf(y[i] | mu[i]);
        }
        return check_term;
    }
}

model{
    /*improper prior on theta in (-inf,inf)*/
    /* directly update the log-probability for sampling */
    target += wt_bin_lpmf(y | mu, weights, n);
}
Stan File: survey weighted quantile regression with splines

functions{
real penalize_spline_lpdf(vector theta, matrix Q,
real tau_theta, int num_bases, int degree) {
    return 0.5 * ( (num_bases-degree) * log(tau_theta) -
    tau_theta * quad_form(Q, theta) ); } 
real rho_p(real p, real u){
    return .5 * (fabs(u) + (2*p - 1)*u); }
real ald_lpdf(vector y, vector mu, vector weights, real tau, real p, int n){
    real w_tot;
    real log_terms;
    real check_term;
    w_tot = sum( weights );
    log_terms = w_tot * (log(tau) + log(p) + log(1-p));
    check_term = 0.0;
    for( i in 1:n )
    {
        check_term = check_term + weights[i] * rho_p( p, (y[i]-mu[i]) );
    }
    check_term = tau * check_term;
    return log_terms - check_term; }}
Stan File: survey weighted quantile regression with splines

model{
    tau_theta ~ gamma( 1.0, 1.0 );
    tau ~ gamma( 1.0, 1.0 );
    theta ~ penalize_spline(Q, tau_theta, num_knots+degree, degree);
    /* directly update the log-probability for sampling */
    target += ald_lpdf(y | mu, weights, tau, p, n);
}
ggplot2

“ggplot2 is a system for declaratively creating graphics, based on *The Grammar of Graphics* (Wilkinson, 2006).”
https://ggplot2.tidyverse.org/

We use the R package ggplot2 (Wickham, 2016) for

- trend lines and ribbons
- violin plots
- heatmaps
- scatter plots with density ellipses
- facetted versions of above
ggplot2: Example with trend lines and ribbons

- **main commands**: `ggplot()`, `+`, `geom_line`, `geom_ribbon`, ...
- **arguments/options**: `data`, `aes “aesthetic”`, ...
- **sub arguments/options**: `x`, `y`, ...

```r
p.t = ggplot() + geom_line(data=data_plot1, aes(x = x, y = mu_W2STGSP), colour = "red", linetype = 1) + geom_ribbon(data=data_plot1, aes(ymin=lo_W2STGSP, ymax=hi_W2STGSP, x = x), alpha=0.1, fill = "red") + labs(x = "age", y = expression(mu)) + theme(legend.position="none") #end of object print(p.t)
```
Example 1: Sampling and Analysing Spouse Pairs

Let $\delta_i$ and $\delta_j$ be indicators that individuals $i$ and $j$ are in the sample. Then the joint indicator $\delta_{ij} = \delta_i \delta_j$.

- **Marginal weight** $w_i = \delta_i / P\{\delta_i = 1\}$
- **Pairwise weight** $\tilde{w}_i = \sum_{i \neq j \in D} (\delta_{ij} / P\{\delta_{ij} = 1\}) / (N_D - 1)$
- For spouses, $N_D = 2$, so ‘multiplicity’ $(N_D - 1) = 1$.
- For marginal models (anyone with a spouse), use $w_i$
- For conditional models (both spouses in the sample), use $\tilde{w}_i$
ggplot2: Comparing Conditional Behaviors of Spouses by Age

Six sets of `geo_line()` and `geo.ribbon()` added as layers via +

- Median alcohol use (days in past month)
- By Age
- By Use of Spouse
  - solid: spouse $\geq 1$
  - dash: spouse = 0
- Compare Weights
  - equal, marginal, pairwise
ggplot2: Comparing Distributions of Alternative Weights

Violin density plots `geom_violin()` of two pairwise weights across simulation size and subpopulation settings via `facet_grid()`
Example 2: Sampling Induced Dependence

▸ Sampling in Practice
  ▸ Unequal probabilities, stratification, and clustering are all incorporated.
  ▸ Individual units aren’t assumed to be sampled independently in general ($\pi_{ij} \neq \pi_i \pi_j$).

▸ Multistage, cluster sampling design
  ▸ Early stages defined by geography: Select dwelling units (DU’s) nested within census block groups (PSU’s)
  ▸ Geographic units stratified ‘implicitly’ via sorting on frame indicators and selected proportional to size measure (Systematic PPS).
  ▸ DU’s selected within segment via random starting point and selecting every $k^{th}$ unit (Systematic)
  ▸ Individuals selected, to exclusion of others in DU (Dependent selection)
ggplot2: Visualizing Sampling Dependence for two PSUs

Heatmap of $\frac{\pi_{ij}}{(\pi_i \pi_j)} - 1$ matrix via `geom_tile()` with custom color scale via `scale_fill_gradient2()`. NA values left empty (gray).
Example 3: Adjusting Coverage of Pseudo Posterior Samples

- \( \hat{\theta}_m \equiv \text{sample pseudo posterior for } m = 1, \ldots, M \text{ draws with mean } \bar{\theta} \)
- \( \hat{\theta}^a_m = \left( \hat{\theta}_m - \bar{\theta} \right) R_2^{-1} R_1 + \bar{\theta} \)
- where \( R_1' R_1 = H_{\theta_0}^{-1} J_{\theta_0}^w H_{\theta_0}^{-1} \), the asy. var. of the pseudo MLE
- \( R_2' R_2 = H_{\theta_0}^{-1} \), the asy. var. of the pseudo posterior (and the MLE under SRS)
- Comparing \( H_{\theta_0}^{-1} J_{\theta_0}^w H_{\theta_0}^{-1} \) to \( H_{\theta_0}^{-1} \) via \( R_2^{-1} R_1 \) captures a multivariate, parameter specific ‘design effect’.
ggplot2: MCMC Samples ($\hat{\theta}_m$, $\hat{\theta}_m^a$) across Survey Designs

Scatter plot `geom_point()`, ellipticals `stat_ellipse()`, comparison group `aes()` options `color =` and `shape =`, across 6 designs `facet_wrap()`

**DE1** One stage DE = 1
**DE5** One stage DE = 5
**PPS1** One Stage PPS
**PPS3** Three Stage PPS
**SPPS1** Stratified PPS1
**SPPS3** Stratified PPS3
tidy-ing up

- More about **Stan**: [http://mc-stan.org/](http://mc-stan.org/)
- More about **ggplot2**: [https://ggplot2.tidyverse.org/](https://ggplot2.tidyverse.org/)
- Other useful tools/packages
  - *survey* and *sampling* packages in R
  - *deriv* function in R and *autodiff* C++ library in Stan
- Future work? R package/wrappers for these models and output.
**URL:** https://arxiv.org/abs/1710.00019


**URL:** http://mc-stan.org/

**URL:** http://ggplot2.org


**URL:** https://arxiv.org/abs/1807.05066