Supplementary Material to Bayesian Data Synthesis and Disclosure Risk Quantification: An Application to the Consumer Expenditure Surveys

Jingchen Hu* and Terrance D. Savitsky†

August 13, 2019

Abstract

This supplement contains: 1. The list of 40 patterns in the CE sample; 2. The full set of within pattern distribution plots of the county label synthesized from the DPMPM, DP-areal synthesizers and the original data distribution; 3. A histogram of identification disclosure risks for the DPMPM, DP-areal and Maximum (from the original data) for the case where only gender and county label are known by the intruder; 4. The Stan script to implement the DP-areal synthesizer.

keywords: Data privacy protection, Disclosure risks, Identification risks, Attribute risks, Synthetic data, Bayesian hierarchical models

*Vassar College, Box 27, 124 Raymond Ave, Poughkeepsie, NY 12604, jihu@vassar.edu.

†U.S. Bureau of Labor Statistics, Office of Survey Methods Research, Suite 5930, 2 Massachusetts Ave NE Washington, DC 20212, Savitsky.Terrance@bls.gov.

Table 1: List of 40 patterns: the index and the number of observations in each pattern.

Index	Observations	Index	Observations
1	27	21	33
2	170	22	229
3	168	23	222
4	194	24	333
5	48	25	128
6	3	26	9
7	193	27	250
8	183	28	254
9	242	29	308
10	61	30	53
11	3	31	8
12	291	32	244
13	275	33	312
14	199	34	184
15	19	35	18
16	4	36	3
17	239	37	198
18	454	38	344
19	169	39	122
20	4	40	10

1 List of 40 Patterns

Table 1 lists the 40 patterns in the CE sample.

2 Within Pattern Density Plots of County Labels among the Synthesizers

Figure 1 to Figure 9 are within pattern distribution plots of the county label synthesized from the DPMPM, DP-areal synthesizers and the original data distribution, from Pattern 5 to Pattern 40.

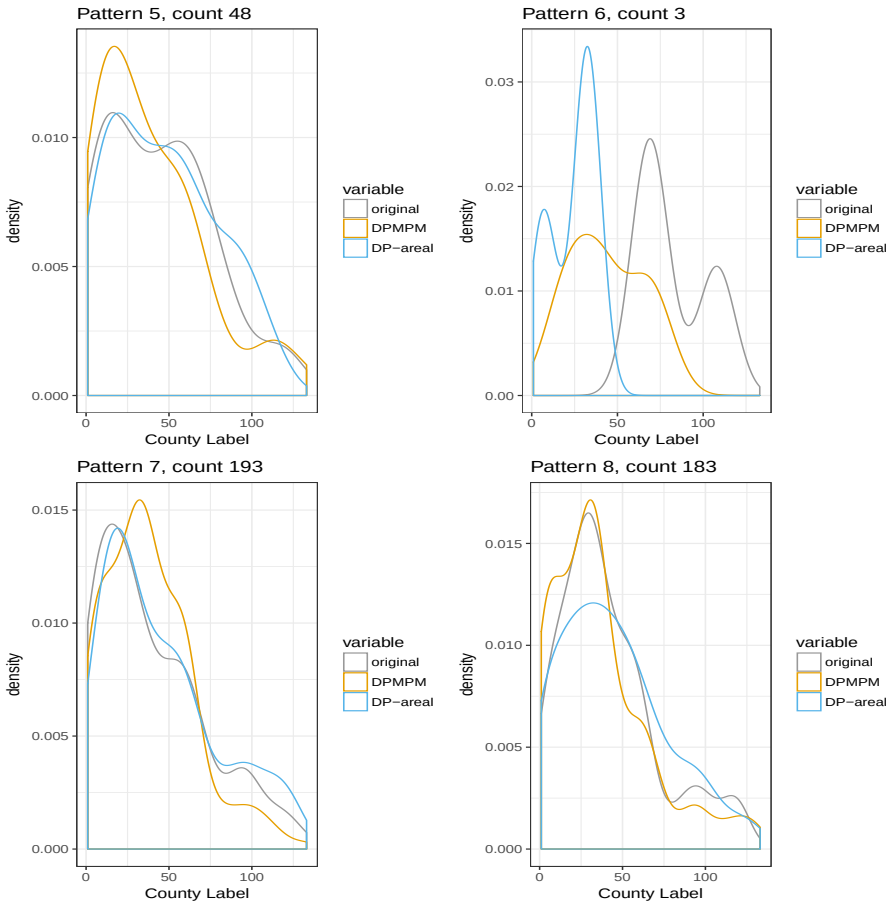

Figure 1: Counties in Pattern 5 to Pattern 8.

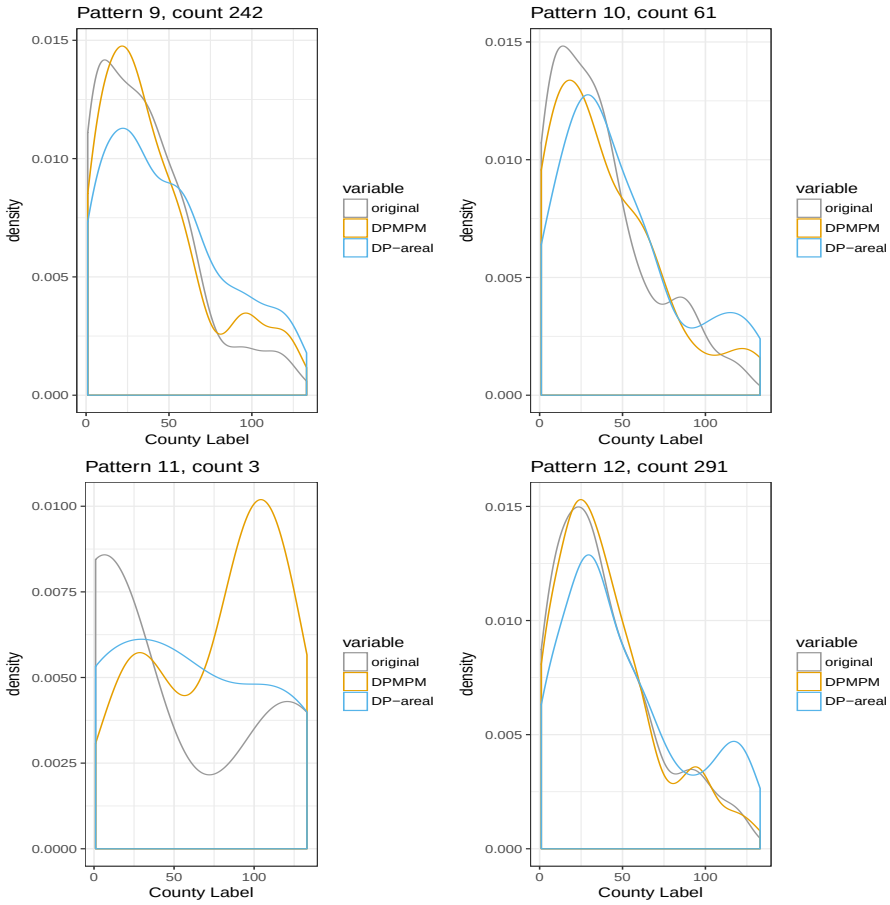

Figure 2: Counties in Pattern 9 to Pattern 12.

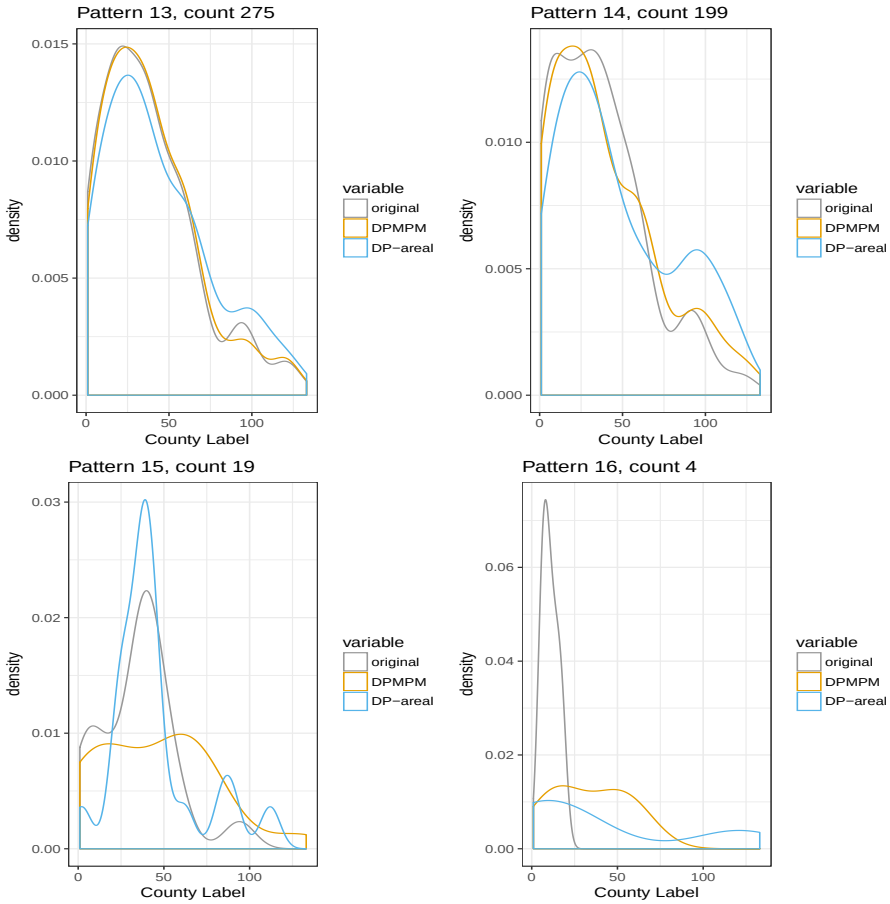

Figure 3: Counties in Pattern 13 to Pattern 16.

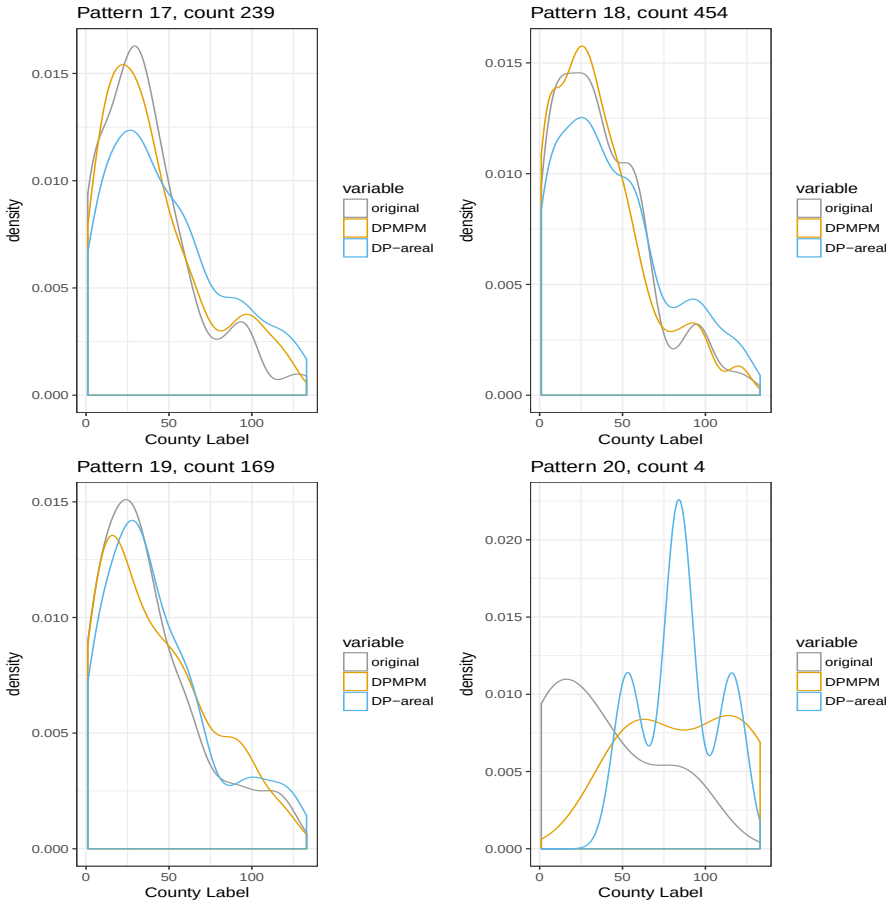

Figure 4: Counties in Pattern 17 to Pattern 20.

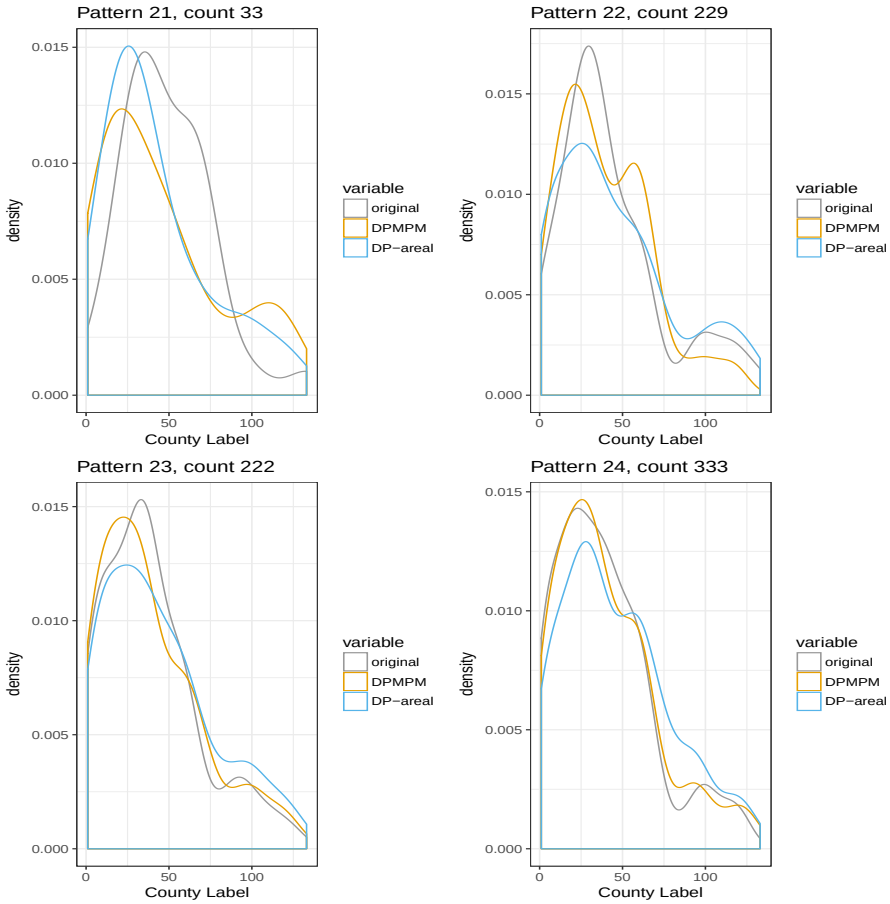

Figure 5: Counties in Pattern 21 to Pattern 24.

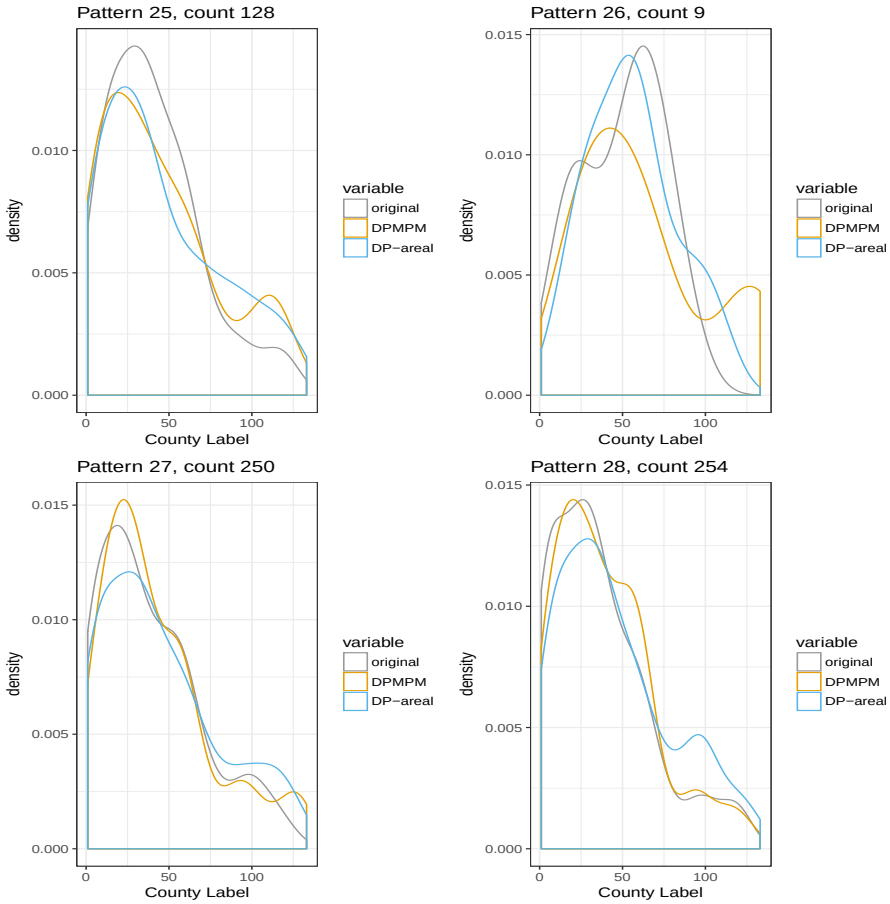

Figure 6: Counties in Pattern 25 to Pattern 28.

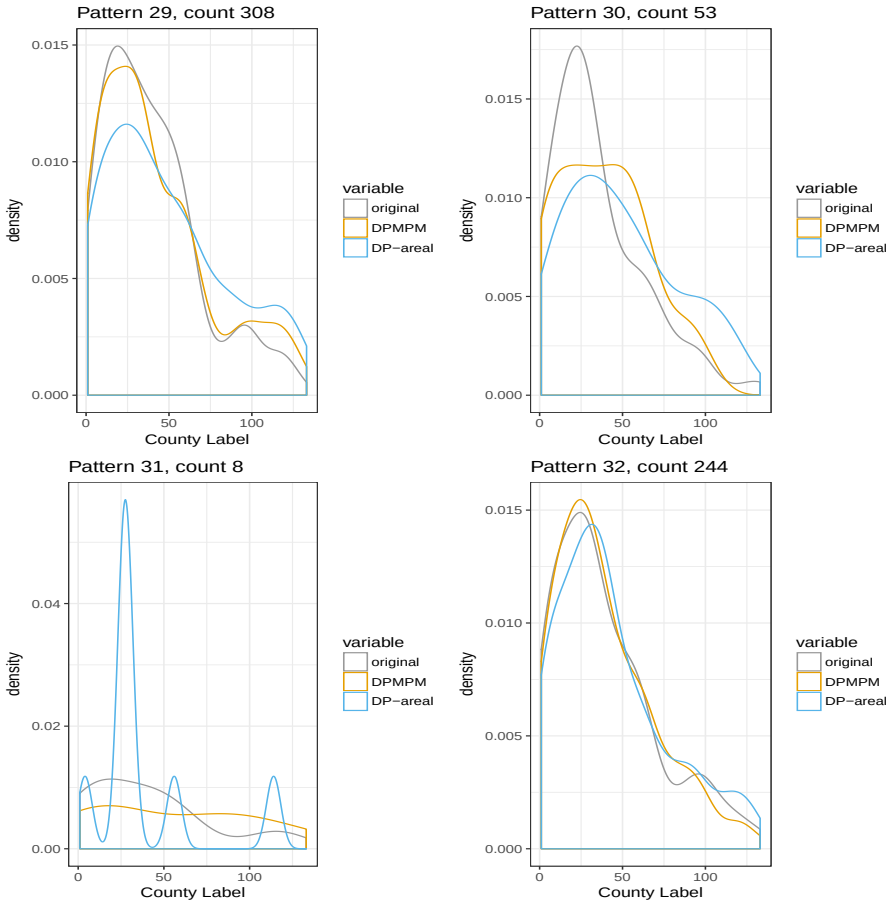

Figure 7: Counties in Pattern 29 to Pattern 32.

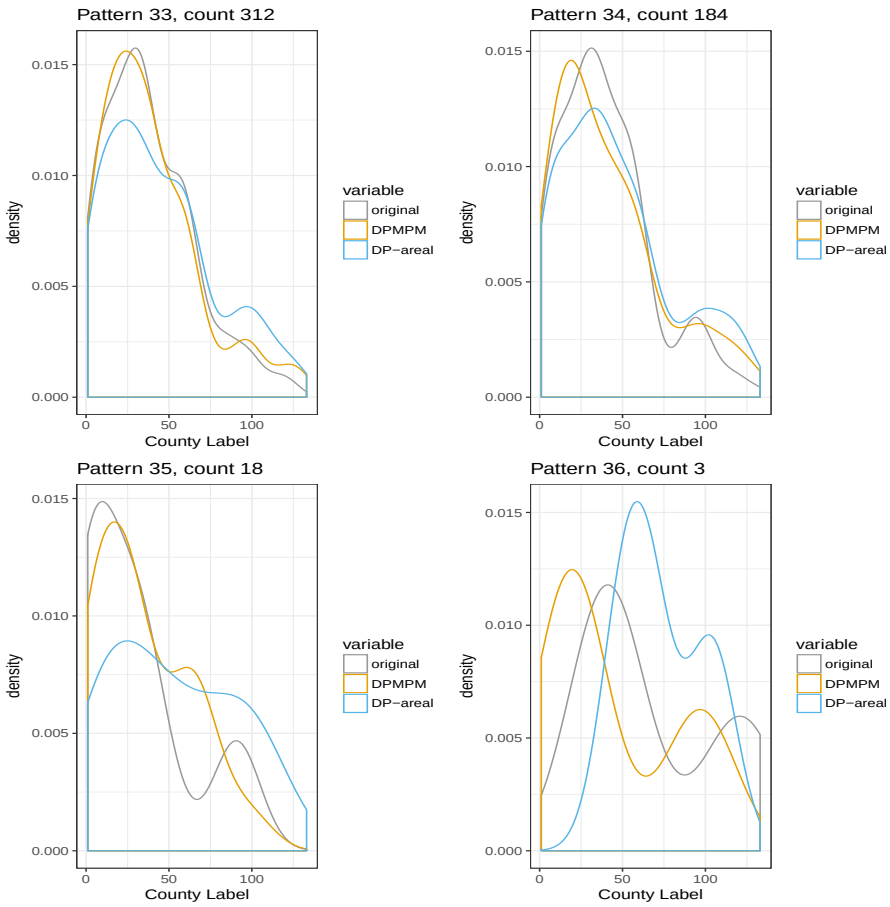

Figure 8: Counties in Pattern 33 to Pattern 36.

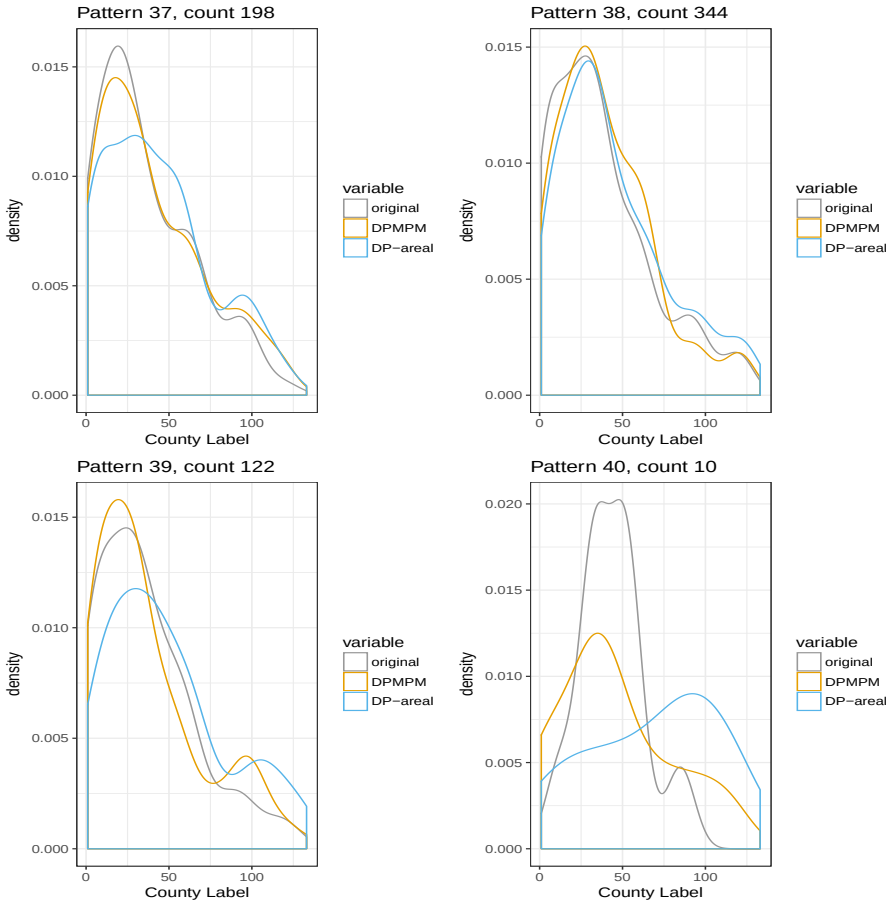

Figure 9: Counties in Pattern 37 to Pattern 40.

3 Identification Disclosure Risk Comparisons under Partially Observed Patterns

Figure 10 is the set of of identification disclosure risks for the DPMPM, DP-areal and Maximum (from the original data) for the case where only gender and county label are known by the intruder.

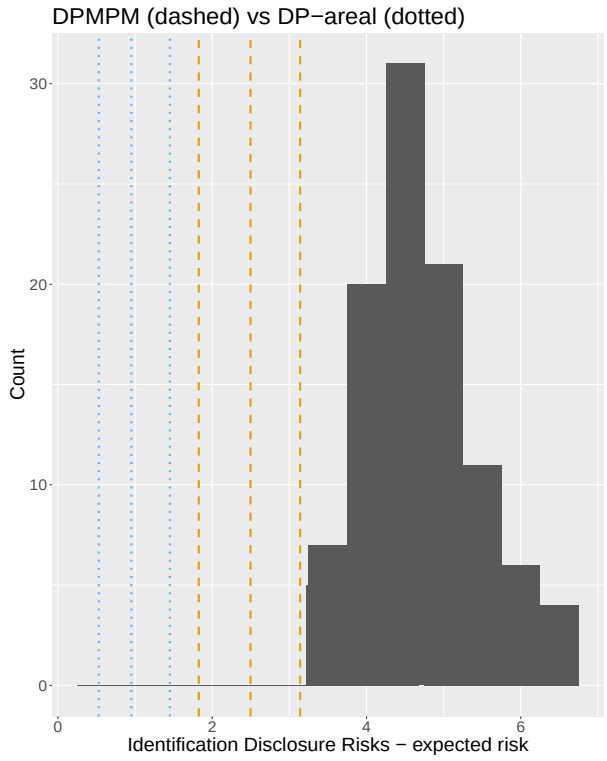

Figure 10: Histogram of expected risks under the maximum risk scenario. Vertical lines include the min, mean, and max among the $m = 20$ synthetic datasets; dashed and orange for DPMPM, and dotted and blue for DP-areal. Known variables: gender and county label.

4 Stan Script to Implement the DP-areal Synthesizer

The following `Stan` script implements the DP-areal synthesizer:

```
functions{

  real normalmix_lpdf(vector log_lambda, vector pi_prob, real mu, vector theta,
   matrix phi,  matrix X, real tau_lambda, int N, int K){
    real log_post;
    log_post = 0;
    for( i in 1:N ) /* by row of N x (R+1) */
    {
      vector[K] ps;
      for( k in 1:K )
      {
        ps[k]  = log(pi_prob[k]) + normal_lpdf(log_lambda[i]| mu + theta[k] +
                                                dot_product(phi[k],X[i]),
                                                 inv(sqrt(tau_lambda)));

      } /* end loop k over clusters / mixture components */
        log_post += log_sum_exp(ps);
    } /* end loop i over N observations */

      return log_post;
  } /* end function normalmix_lpdf() */

} /* end function{} block */


  data{
```

```
  int<lower=1> N; // number of unique combinations of all attributes
  int<lower=1> K; // number of clusters
  int<lower=1> p; // number of non-geographic attributes
  row_vector[p] dj; // vector storing the number of levels of p non-geographic attributes
  int<lower=1> R; // number of total attribute levels: sum(dk)
  matrix[N,R] X; // each row is an R-by-1 vector comprising a one at position x_{ir}^{(b)}
  int c[N]; // set of N observations: the counts
} /* end data block */


transformed data{
  vector<lower=0>[K] ones_K;
  ones_K =  rep_vector(1,K); /* dirichlet prior on alpha has equal shapes */
} /* end transformed parameters block */


parameters{
  vector[N] log_lambda; /* poisson rates */
  real mu; /* global intercept */
  real alpha; /* DP concentration parameter on mixture model for point estimate */
  matrix[K,R] phi;
  vector[K] theta;
  simplex[K] pi_prob; /* mixture probabilities */
  vector[R] mu_phi;
  real<lower=0> tau_theta;
  real<lower=0> tau_phi;
  vector<lower=0>[R] sigma_phi;
  cholesky_factor_corr[R] L_phi;
  real<lower=0> tau_mu; /* precision in prior for mu */
  real<lower=0> tau_lambda; /* precision in prior for log_lambda[i] */
```

```
} /* end parameters block */


transformed parameters{
  vector[N] lambda; /* fitted values */

    for( i in 1:N )
    {
      lambda[i] = exp(log_lambda[i]);
    } /* end loop i over domains */


} /* end transformed parameters block */

model{

  // priors for cluster locations
  alpha           ~ gamma( 1.0, 1.0 ); /* DP concentration parameter */
  pi_prob         ~ dirichlet( alpha/K * ones_K ); /* instantiate a truncated DP prior */

    // normal prior for K x 1, theta
  theta           ~ normal(0,inv(sqrt(tau_theta))); /* vectorized */
  tau_theta       ~ gamma( 1.0, 1.0 );

  // multivariate Gaussian prior for R x 1, phi[k,]
  mu_phi          ~ normal(0,inv(sqrt(tau_phi))); /* vectorized */
  tau_phi         ~ gamma( 1.0, 1.0 );
  L_phi           ~ lkj_corr_cholesky(4);
  sigma_phi       ~ student_t(3,0,1); /* vectorized */
    for(k in 1:K )
    {
```

```
      /* phi[k] is the kth row of K x R, phi */
        to_vector(phi[k]) ~ multi_normal_cholesky(mu_phi,
                                                  diag_pre_multiply(sigma_phi,L_phi));
    }

  mu              ~ normal(0,inv(sqrt(tau_mu)));
  tau_mu          ~ gamma( 1.0, 1.0 );

  // latent response (mean) likelihood on the log scale - mixture of normals prior
  // note that the normal prior allows for over-dispersion
  log_lambda      ~ normalmix(pi_prob, mu, theta, phi, X, tau_lambda, N, K);
  tau_lambda      ~ gamma( 1.0, 1.0 );

  // observed response likelihood
  c  ~ poisson_log(log_lambda);


} /* end model{} block */
```