# Appendices for "d-blink: Distributed End-to-End Bayesian Entity Resolution"

Neil G. Marchant[a] Andee Kaplan[b] Daniel N. Elazar[c]
Benjamin I. P. Rubinstein[a] Rebecca C. Steorts[d]

[a]School of Computing and Information Systems, University of Melbourne
[b]Department of Statistics, Colorado State University
[c]Methodology Division, Australian Bureau of Statistics
[d]Department of Statistical Science and Computer Science, Duke University
Principal Mathematical Statistician, United States Census Bureau

September 14, 2020

# A Derivation of the posterior distribution

Here we sketch the derivation of the joint posterior distribution over the unobserved variables conditioned on the observed record attributes $\boldsymbol{X}^{(o)}$, which is given in Equation 10 of the paper. First we read the factorization off the plate diagram in Figure 1, together with the conditional dependence assumptions detailed in Section 3.2 of the paper. We obtain the following expression, up to a normalisation constant:

$$\begin{aligned} p(\boldsymbol{\Gamma}, \boldsymbol{\Lambda}, \boldsymbol{Y}, \boldsymbol{Z}, \boldsymbol{\Theta}, \boldsymbol{X}^{(m)} | \boldsymbol{X}^{(o)}, \boldsymbol{O}) &\propto \prod_{e,a} p(y_{ea}|\phi_a) \times \prod_{t,a} p(\theta_{ta}|\alpha_a, \beta_a) \\ &\times \prod_{t,r} \Big\{ p(\gamma_{tr}|\boldsymbol{Y}) p(\lambda_{tr}|\gamma_{tr}, \boldsymbol{Y}) \prod_a p(z_{tra}|\theta_{ta}) \Big\} \times \prod_{\substack{t,r,a \\ o_{tra}=1}} p(x_{tra}|z_{tra}, \lambda_{tr}, y_{\lambda_{tr}a}) \\ &\times \prod_{\substack{t,r,a \\ o_{tra}=0}} p(x_{tra}|z_{tra}, \lambda_{tr}, y_{\lambda_{tr}a}). \end{aligned}$$

Ideally, we'd like to marginalize out all variables except $\boldsymbol{\Lambda}$ and $\boldsymbol{Y}$ (the variables of interest), however this is not tractable analytically. Fortunately, we can marginalize out the missing record attributes $\boldsymbol{X}^{(m)}$ which yields Equation 10 from the paper:

$$\begin{aligned} p(\boldsymbol{\Gamma}, \boldsymbol{\Lambda}, \boldsymbol{Y}, \boldsymbol{Z}, \boldsymbol{\Theta} | \boldsymbol{X}^{(o)}, \boldsymbol{O}) &\propto \prod_{e,a} p(y_{ea}|\phi_a) \times \prod_{t,a} p(\theta_{ta}|\alpha_a, \beta_a) \\ &\times \prod_{t,r} \Big\{ p(\gamma_{tr}|\boldsymbol{Y}) p(\lambda_{tr}|\gamma_{tr}, \boldsymbol{Y}) \prod_a p(z_{tra}|\theta_{ta}) \Big\} \times \prod_{\substack{t,r,a \\ o_{tra}=1}} p(x_{tra}|z_{tra}, \lambda_{tr}, y_{\lambda_{tr}a}). \end{aligned}$$

We can expand this further by substituting the conditional distributions given in Section 3.2 of the paper. This yields:

$$\begin{aligned} p(\boldsymbol{\Gamma}, \boldsymbol{\Lambda}, \boldsymbol{Y}, \boldsymbol{Z}, \boldsymbol{\Theta} | \boldsymbol{X}^{(o)}, \boldsymbol{O}) &\propto \prod_{e,a} \phi_a(y_{ea}) \times \prod_{t,a} \theta_{ta}^{\alpha_a - 1} (1 - \theta_{ta})^{\beta_a - 1} \\ &\times \prod_{t,r} \Big\{ \mathbb{I}[\lambda_{tr} \in \mathcal{E}_{\gamma_{tr}}(\boldsymbol{Y})] \prod_a \theta_{ta}^{z_{tra}} (1 - \theta_{ta})^{1 - z_{tra}} \Big\} \\ &\times \prod_{\substack{t,r,a \\ o_{tra}=1}} \Big\{ (1 - z_{tra}) \mathbb{I}[x_{tra} = y_{\lambda_{tr}a}] + z_{tra}\, \psi_a(x_{tra}|y_{\lambda_{tr}a}) \Big\}. \end{aligned} \tag{S1}$$

# B Equivalence of `d-blink` and `blink`

In this section, we present proofs of Propositions 1 and 2, which show that the inferences we obtain from `d-blink` are equivalent to those we would obtain from `blink` under certain conditions.

## B.1 Proof of Proposition 1: equivalence of distance/similarity representations

It is straightforward to show that `sim` as defined in Equation 11 of the paper satisfies the requirements of Definition 3.3. All that remains is to show that the two parameterizations of the distortion distribution $\psi_a$ are equivalent. Beginning with $\psi_a$ as parameterized in `blink`, we substitute Equation 11 and observe that

$$\psi_a(v|w) \propto \phi_a(v)\mathrm{e}^{-\texttt{dist}_a(v,w)} = \phi_a(v)\mathrm{e}^{d_{\max;\mathrm{a}}+\texttt{sim}_a(v,w)} \propto \phi_a(v)\mathrm{e}^{\texttt{sim}_a(v,w)}.$$

This is identical to our parameterization in Equation 9. □

## B.2 Proof of Proposition 2: equivalence of `d-blink` and `blink`

Given that

- Proposition 1 holds,
- the distortion hyperparameters are the same for all attributes, and
- all record attributes are observed,

the only factor in the posterior that differs from `blink` is:

$$\prod_{t,r} p(\lambda_{tr}|\gamma_{tr}, \boldsymbol{Y})p(\gamma_{tr}|\boldsymbol{Y}). \tag{S2}$$

Substituting the density for the conditional distributions for a single $t, r$ factor yields:

$$p(\lambda_{tr}|\gamma_{tr}, \boldsymbol{Y})p(\gamma_{tr}|\boldsymbol{Y}) = \frac{\mathbb{I}[\lambda_{tr} \in \mathcal{E}_{\gamma_{tr}}(\boldsymbol{Y})]}{|\mathcal{E}_{\gamma_{tr}}(\boldsymbol{Y})|} \times \frac{|\mathcal{E}_{\gamma_{tr}}(\boldsymbol{Y})|}{E} = \frac{1}{E}\mathbb{I}[\lambda_{tr} \in \mathcal{E}_{\gamma_{tr}}(\boldsymbol{Y})]\,.$$

Putting this in Equation S2 and marginalizing over $\boldsymbol{\Gamma}$ we obtain:

$$\prod_{t,r}\sum_{\gamma_{tr}=1}^{B} p(\lambda_{tr}|\gamma_{tr}, \boldsymbol{Y})p(\gamma_{tr}|\boldsymbol{Y}) = \prod_{t,r}\frac{1}{E}\sum_{\gamma_{tr}=1}^{B}\mathbb{I}[\lambda_{tr} \in \mathcal{E}_{\gamma_{tr}}(\boldsymbol{Y})] = \prod_{t,r}\frac{1}{E}\mathbb{I}[\lambda_{tr} \in \{1,\ldots,E\}]\,,$$

which is the factor that appears in the posterior for `blink`. □

# C Splitting rules for the *k*-d tree blocking function

In Section 4.2 of the paper we outline a blocking function inspired by $k$-d trees. When inserting a node in the tree, we require a splitting rule that partitions the input set of values. In ordinary $k$-d trees, the median is often used for this purpose, however it is not appropriate for the discrete input sets that we encounter. As a result, we propose the following alternative splitting rules:

1. *Ordered median.* This rule is appropriate if the set of input attribute values is large and/or has a natural ordering. If there is no natural ordering, an artificial ordering must be applied (e.g. lexicographic ordering). The splitting rule is determined by sorting the input values and finding the median, accounting for the frequency of each value. Attribute values ordered before (after) the median are passed to the left (right) child node.

2. *Reference set.* This rule is appropriate if the set of input attribute values is small with no natural ordering. The splitting rule is determined by using a first-fit bin-packing algorithm to split the values into two roughly equal-sized bins, accounting for the frequency of each value. One of these bins is then labeled the "reference set". Attribute values (not) in the reference set are passed to the left (right) child node.

# D Gibbs update distributions

Here we list the conditional distributions for the Gibbs updates. These are derived by referring to the posterior distribution in Equation S1.

## D.1 Update for $\theta_{ta}$

$$\theta_{ta}|\boldsymbol{Z}, \boldsymbol{\Lambda}, \boldsymbol{\Gamma}, \boldsymbol{Y}, \boldsymbol{X}^{(o)}, \boldsymbol{O} \sim \text{Beta}[z_{t\cdot a} + \alpha_a, R_t - z_{t\cdot a} + \beta_a] \tag{S3}$$

where $z_{t\cdot a} := \sum_{r=1}^{R_t} z_{tra}$.

## D.2 Update for $z_{tra}$

$$z_{tra}|\boldsymbol{\Lambda}, \boldsymbol{\Gamma}, \boldsymbol{Y}, \boldsymbol{\Theta}, \boldsymbol{X}^{(o)}, \boldsymbol{O} \sim (1 - o_{tra})\,\text{Bernoulli}[\theta_{ta}] + o_{tra}\,\text{Bernoulli}[\zeta_a(\theta_{ta}, x_{tra}, y_{\lambda_{tr}a})] \tag{S4}$$

where $\zeta_a(\theta, x, y) = \begin{cases} 1, & \text{if } x \neq y, \\ \frac{\theta\psi_a(x|y)}{\theta\psi_a(x|y) - \theta + 1}, & \text{otherwise.} \end{cases}$

## D.3 Update for $\lambda_{tr}$

$$p(\lambda_{tr}|\boldsymbol{\Gamma}, \boldsymbol{Y}, \boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{X}^{(o)}, \boldsymbol{O}) \propto \mathbb{I}[\lambda_{tr} \in \mathcal{E}_{\gamma_{tr}}(\boldsymbol{Y})] \prod_{\substack{a \\ o_{tra}=1}} \Big\{(1 - z_{tra})\mathbb{I}[x_{tra} = y_{\lambda_{tr}a}] + z_{tra}\psi_a(x_{tra}|y_{\lambda_{tr}a})\Big\}. \tag{S5}$$

# E Perturbation sampling algorithm

In Proposition 3 of the paper, we show how to express a target pmf $p$ (from which we'd like to draw random variates) as a mixture over a base pmf $q$ and a perturbation pmf $v$. Algorithm S1 demonstrates how to efficiently draw random variates from the target pmf using this mixture representation.

**Algorithm S1** Perturbation sampling for $p(x|\omega)$

**Input:** map from $x, \omega \in \mathcal{X}^\star \times \Omega \to \epsilon(x|\omega)$; map from $x \in \mathcal{X} \to q(x)$; pre-initialized Alias sampler for $q$.

1: $v \leftarrow \emptyset$ ▷ empty map
2: **for** $x \in \mathcal{X}^\star$ **do**
3: $\quad v(x) \leftarrow q(x)\epsilon(x|\omega)$
4: **end for**
5: $c \leftarrow 1/\sum_{x \in \mathcal{X}^\star} v(x)$ ▷ normalization
6: $s \sim \text{Bernoulli}\left[\frac{c}{1+c}\right]$
7: **if** $s = 1$ **then**
8: $\quad$ **Return:** $x \sim q(\cdot)$ ▷ using input Alias sampler
9: **else**
10: $\quad v \leftarrow c \cdot v$
11: $\quad$ **Return:** $x \sim v(\cdot)$ ▷ using new Alias sampler
12: **end if**

### E.1 Proof of Proposition 4: complexity of perturbation sampling

Let us analyze the time complexity of Algorithm S1. Lines 2–6 are $O(|\mathcal{X}^\star|)$. By properties of the Alias sampler (Vose, 1991), line 8 is $O(1)$ and line 11 is $O(|\mathcal{X}^\star|)$. Thus the overall complexity is $O(|\mathcal{X}^\star|)$.

## F Further details on the experimental set-up

### F.1 Data sets

We provide a brief description of each data set below. All data sets come with some form of "ground truth", which we use for evaluation purposes. However, the ground truth for `NCVR` and `SHIW0810` (two of the real data sets) may not be error-free as indicated below.

- `ABSEmployee`. A synthetic data set used internally for linkage experiments at the Australian Bureau of Statistics. It simulates an employment census and two supplementary surveys (it is not derived from any real data sources). We used four categorical attributes: `MB`, `BDAY`, `BYEAR` and `SEX`.
- `NCVR`. Two snapshots from the North Carolina Voter Registration database taken two months apart (Christen, 2014). The snapshots are filtered to include only those voters whose details changed over the two-month period. We used `first_name`, `middle_name` and `last_name` as string-type attributes; and `age`, `gender` and `zip_code` as categorical attributes. Unique voter identifiers are provided, however they are known to contain some errors (Christen, 2014).
- `NLTCS`. A subset of the U.S. National Long-Term Care Survey (Manton, 2010) comprising the 1982, 1989 and 1994 waves. It was necessary to use a subset, as race

was subsampled in the other three years, making it unsuitable for ER. We used four categorical attributes: `SEX`, `DOB`, `STATE` and `REGOFF`. Unique identifiers are available which are known to be of high quality.

- `SHIW0810`. A subset from the Bank of Italy's Survey on Household Income and Wealth (Banca d'Italia, n.d.) comprising the 2008 and 2010 waves. We used eight categorical attributes: `IREG`, `SESSO`, `ANASC`, `STUDIO`, `PAR`, `STACIV`, `PERC` and `CFDIC`. Unique identifiers were inferred using a deterministic algorithm, which may not be error-free. Further information and open-source code is provided at `http://github.com/ngmarchant/shiw`.

- `RLdata10000`. A synthetic data set provided with the `RecordLinkage` R package (Sariyar and Borg, 2010). We used `fname_c1` and `lname_c1` as string-type attributes and `bd`, `bm`, `by` as categorical attributes. The `fname_c2` and `lname_c2` were excluded as they have a high fraction of missing values.

## F.2 Implementation and hardware

Our implementation of `d-blink` is written in Scala and depends on Apache Spark 2.3.1 (a distributed computing framework). Since `d-blink` requires control over the partitioning (entities and linked records *must* reside on their assigned partitions), we used the RDD API with a custom partitioner. Our custom-built server ran in local (pseudo-cluster) mode, with 2× 28-core Intel Xeon Platinum 8180M CPUs for a total of 112 threads (with HyperThreading); and 128GB of allocated RAM on the driver.

## F.3 Parameter settings and initialization

We used the following parameter settings for all experiments.

- The distortion hyperparameters $\alpha_a, \beta_a$ were set to encode a prior mean distortion probability of approximately 1%, with the strength varying in proportion to the total number of records $R$:

$$\alpha_a = R \times 10\% \times 1\% \text{ and } \beta_a = R \times 10\% \text{ for all } a.$$

- The size of the latent entity population $E$ was set to $R$. This corresponds to a prior mean number of observed entities of $(1 - \mathrm{e}^{-1})R \approx 0.63R$, as shown by Steorts et al. (2016). It is important not to set $E$ too low, as it places an upper bound on the number of entities present in the data.

- The entity attribute distributions $\{\phi_a\}$ were set empirically based on the observed record attributes. Specifically, we set

$$\phi_a(v) = \frac{\sum_{t=1}^{T} \sum_{r=1}^{R_t} o_{tra} \mathbb{I}[x_{tra} = v]}{\sum_{t=1}^{T} \sum_{r=1}^{R_t} o_{tra}} \quad \text{for all } a.$$

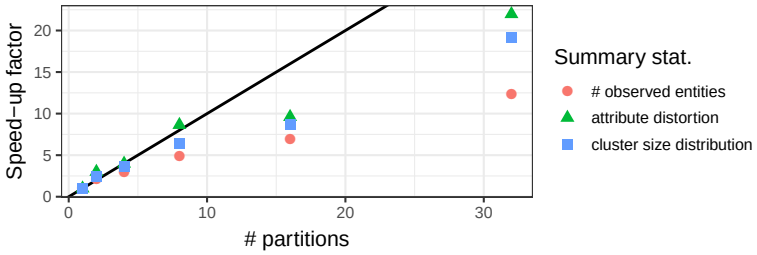


Figure S1: Efficiency of `d-blink` as a function of the number of blocks $B$ and summary statistic of interest (larger is better). The speed-up measures the ESS rate relative to the ESS rate for $B = 1$ (no partitioning) for the `NLTCS` data set.

- For simplicity, we treated all attributes as either "categorical-type" with similarity function $\texttt{sim}_{\text{const}}$ or "string-type" with similarity function $10.0 \times \texttt{sim}_{\text{nEd}}$ (these are defined in Section 3.3).
- The similarity cut-off for string-type attributes was set to 7.0, following advice in the `RecordLinkage` R package (Sariyar and Borg, 2010).
- We used the $k$-d tree blocking function as defined in Section 4.2. The *reference set* splitting rule was used for input sets with 30 or fewer elements—the *ordered median* splitting rule was used otherwise.

To initialize the Markov chain, we linked each record to a unique entity and copied the record attributes into the entity attributes, assuming no distortion. Any entity attributes that were missing after this process (due to missing record attributes) were filled by drawing an attribute value from the empirical distribution. We set the thinning interval to 10—i.e. we only saved every tenth step along the chain. This increases the effective sample size for a given storage budget.

# G Results on Amazon EC2

We repeated two of the experiments described in Section 7.1 of the main paper on a cluster running in the Amazon Elastic Compute Cloud (EC2). For the worker (executor) nodes, we used varying numbers of `m5.xlarge` instances with 4 vCores, 16 GiB memory and 32 GiB of Elastic Block Store (EBS) storage. Due to the increased latency and decreased bandwidth between the compute nodes, we expected the efficiency to decrease. This is indeed what we observed.

Figure S1 plots the speed-up as a function of the number of blocks $B$ relative to a baseline with no partitioning. We observe poorer scaling with $B$ compared to the results we obtained on our local server (c.f. Figure 5 in the main paper). Figure S2 plots the efficiency as a function of the sampling method with $B = 16$. The results are qualitatively similar to the ones we obtained using our local server (c.f. Figure 6 in the main paper). However, the ESS rate was reduced for all samplers as expected due to increased communication costs.

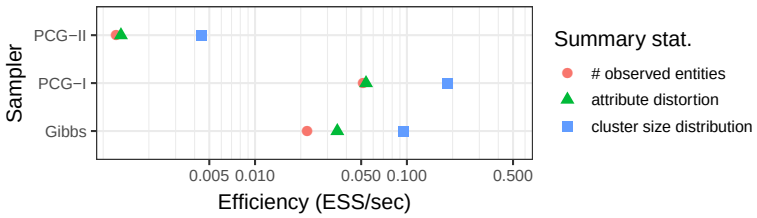


Figure S2: Efficiency of `d-blink` as a function of the sampler and summary statistic of interest (larger is better). All measurements are for the `NLTCS` data set with $B = 16$.

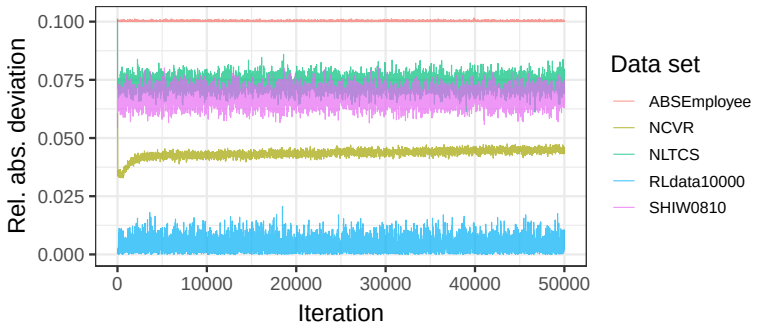


Figure S3: Balance of the blocks for a single run on each data set. The balance is measured in terms of the relative absolute deviation from the perfectly balanced configuration. The number of blocks $B = 64, 64, 16, 2, 8$ for each data set (in the order listed in the legend).

# H Balance of the blocks

In Section 4.2, we proposed a blocking function based on $k$-d trees, and argued that it could yield balanced blocks with good entity separation. While running `d-blink` with the $k$-d tree blocking function, we recorded the size of the blocks ($|\mathcal{E}_b|$ for all $b$) to assess whether they were well-balanced. Figure S3 illustrates the results in terms of the relative absolute deviation from the perfectly balanced configuration (where the entities are divided equally among the blocks). We can see that the $k$-d tree partitioner is functioning quite well—the deviation from the perfectly balanced configuration is no more than 10% for all data sets.

# I Uncertainty measures

`d-blink` allows for measures of uncertainty to be reported, unlike the baseline methods, since we have the full posterior distribution. For example, in Figure S4 we compute posterior estimates for the number of entities present in each data set, with 95% Bayesian credible intervals. Note that the posterior estimates are typically quite sharp. This seems to confirm arguments by Steorts et al. (2016) regarding the informativeness of the prior for the linkage structure in `blink`. Research on less informative priors is ongoing (Zanella et al., 2016).

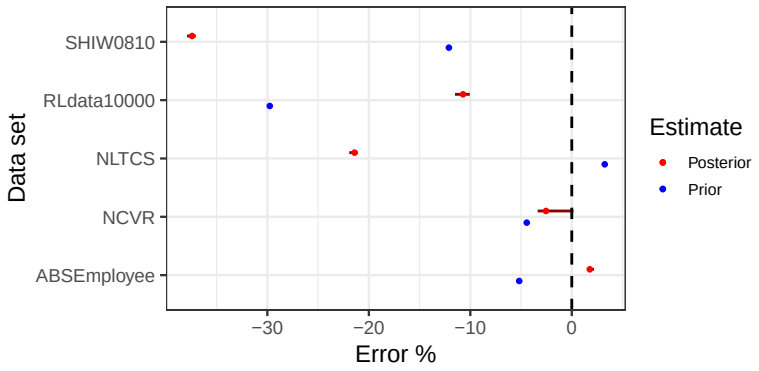


Figure S4: Percentage error in the posterior/prior estimates for the number of observed entities for `d-blink`. The posterior estimates are generally sharp and underestimate the true number of observed entities.

# J Sensitivity analysis

We conducted an empirical sensitivity analysis for `d-blink` using the `RLdata10000` data set. We selected this data set as it is relatively small, which made it quick to run the inference for various hyperparameter combinations. The parameters tested were:

- $\alpha_\ell, \beta_\ell$: the shape parameters for the Beta prior on the distortion probabilities. We used the same values for all attributes ($a$).
- $E$: the size of the latent population.
- $s_{\max}$: the scaling factor for the similarity function. This controls the inverse temperature of the softmax distribution for the distorted attribute values.

We varied each of these parameters in turn, while holding all other parameters fixed. For the Beta prior on the distortion probabilities, we first varied the strength while fixing the prior mean to $\sim 1\%$, then we varied the mean (1%, 5% and 10%) while fixing $\alpha + \beta$ (related to the strength). Table S1 presents the evaluation measures for each combination of parameters. The results indicate that the inferred linkage structure is relatively sensitive to all of the parameters, however sensitivity is in general predictable, following clear and intuitive trends. Of particular interest is the fact that the model performs best when the Beta prior on the distortion probabilities is sharply peaked near zero. It seems that the model has a tendency to overestimate the amount of distortion, particularly in the absence of ground truth.

# K Details of inference for the case study to the 2010 Decennial Census

We ran inference for 15,000 iterations using the PCG-I sampler. After removing 5,000 iterations as burn-in and applying thinning with an interval of 10, we obtained 1,000 approximate samples from the posterior. Convergence diagnostics are consistent with those

Table S1: Sensitivity analysis for various parameters combinations using `RLdata10000`. The first group of rows tests the effect of varying the *strength* of the Beta prior, the second group tests the effect of varying the *mean* of the Beta prior, the third group tests the effect of varying the population size, and the fourth group tests the effect of varying the scaling factor for the similarity function.

| Distortion | | Pop. size | Max. sim. | Pairwise measures | | | Cluster measures | |
|---|---|---|---|---|---|---|---|---|
| $\alpha$ | $\beta$ | $E$ | $s_{\max}$ | Precision | Recall | F1-score | ARI | Err. # clust. |
| **0.1** | **10.0** | 10000 | 10.0 | 0.5342 | 0.9990 | 0.6962 | 0.6962 | −17.47% |
| **1.0** | **100.0** | 10000 | 10.0 | 0.5435 | 0.9990 | 0.7040 | 0.7040 | −16.58% |
| **10.0** | **1000.0** | 10000 | 10.0 | 0.6334 | 0.9970 | 0.7747 | 0.7747 | −10.97% |
| **100.0** | **10000.0** | 10000 | 10.0 | 0.9180 | 0.9850 | 0.9503 | 0.9503 | −1.595% |
| **10.0** | **1000.0** | 10000 | 10.0 | 0.6334 | 0.9970 | 0.7747 | 0.7747 | −10.97% |
| **50.5** | **959.5** | 10000 | 10.0 | 0.6132 | 0.9970 | 0.7593 | 0.7593 | −11.90% |
| **101.0** | **909.0** | 10000 | 10.0 | 0.5992 | 0.9970 | 0.7485 | 0.7485 | −12.90% |
| 10.0 | 1000.0 | **9000** | 10.0 | 0.5306 | 0.9970 | 0.6926 | 0.6926 | −15.65% |
| 10.0 | 1000.0 | **10000** | 10.0 | 0.6334 | 0.9970 | 0.7747 | 0.7747 | −10.97% |
| 10.0 | 1000.0 | **11000** | 10.0 | 0.6999 | 0.9960 | 0.8221 | 0.8221 | −7.365% |
| 10.0 | 1000.0 | 10000 | **5.0** | 0.6927 | 0.9940 | 0.8164 | 0.8164 | −22.12% |
| 10.0 | 1000.0 | 10000 | **10.0** | 0.6334 | 0.9970 | 0.7747 | 0.7747 | −10.97% |
| 10.0 | 1000.0 | 10000 | **50.0** | 0.2112 | 0.3920 | 0.2745 | 0.2745 | −12.50% |

reported for the other data sets in Appendix L, and are complicated to release due to the fact that the data is protected under Title 13. Releasing each iteration of a Gibbs sampler could potentially say something about individuals in the population, and thus, for privacy reasons, these diagnostics are omitted.

# L Trace plots

## L.1 Attribute-level distortion

The following figures relate to the aggregate distortion per attribute for each data set. On the left are the trace plots, which show the aggregate distortion for each attribute (stacked vertically) along the Markov chain. On the right are the corresponding autocorrelation plots.

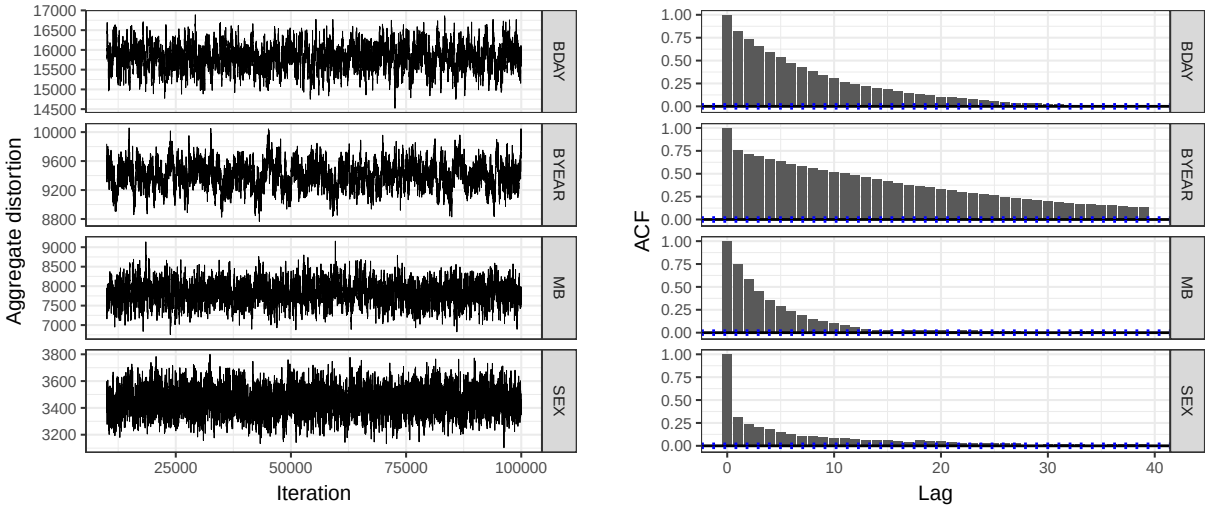


Figure S5: Attribute-level distortion for `ABSEmployee`

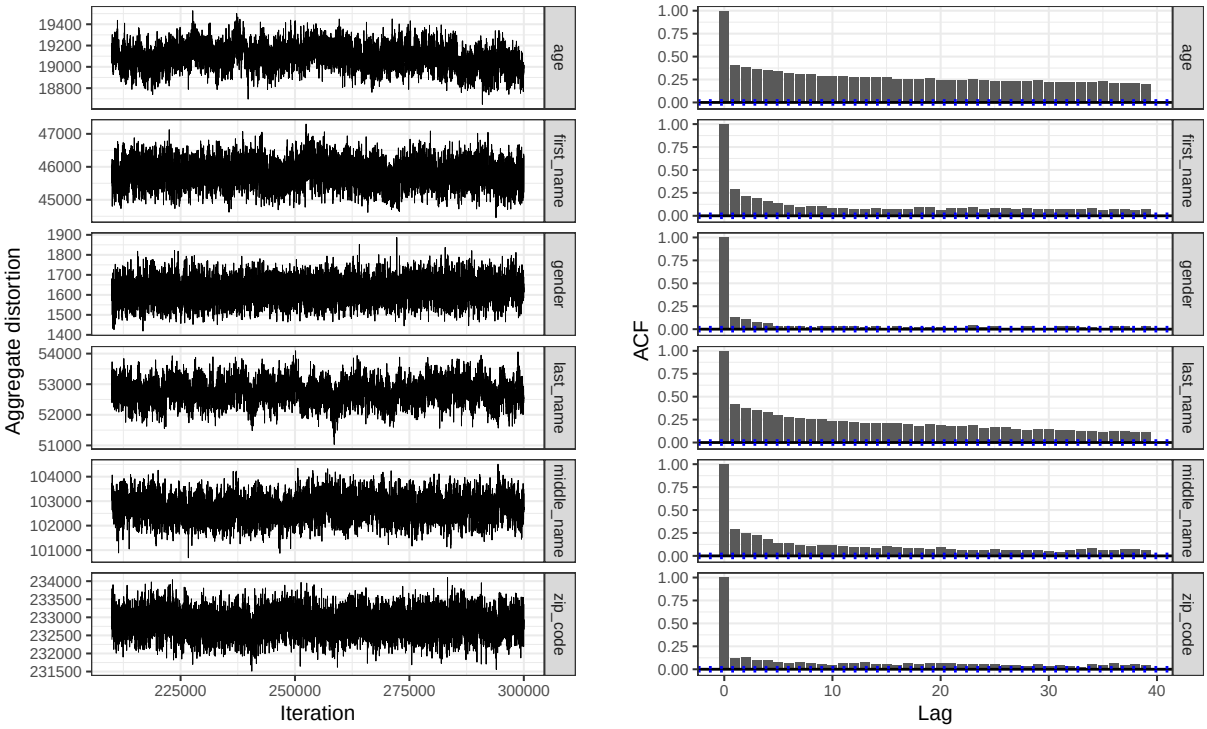


Figure S6: Attribute-level distortion for `NCVR`

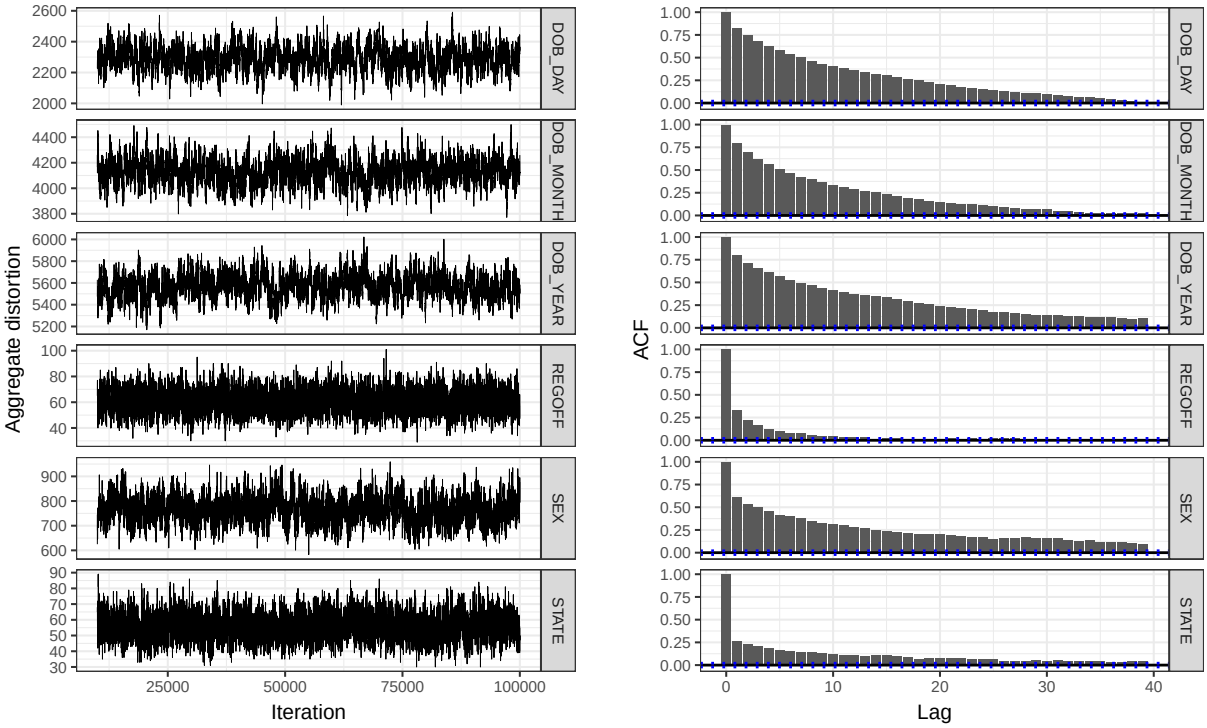


Figure S7: Attribute-level distortion for `NLTCS`

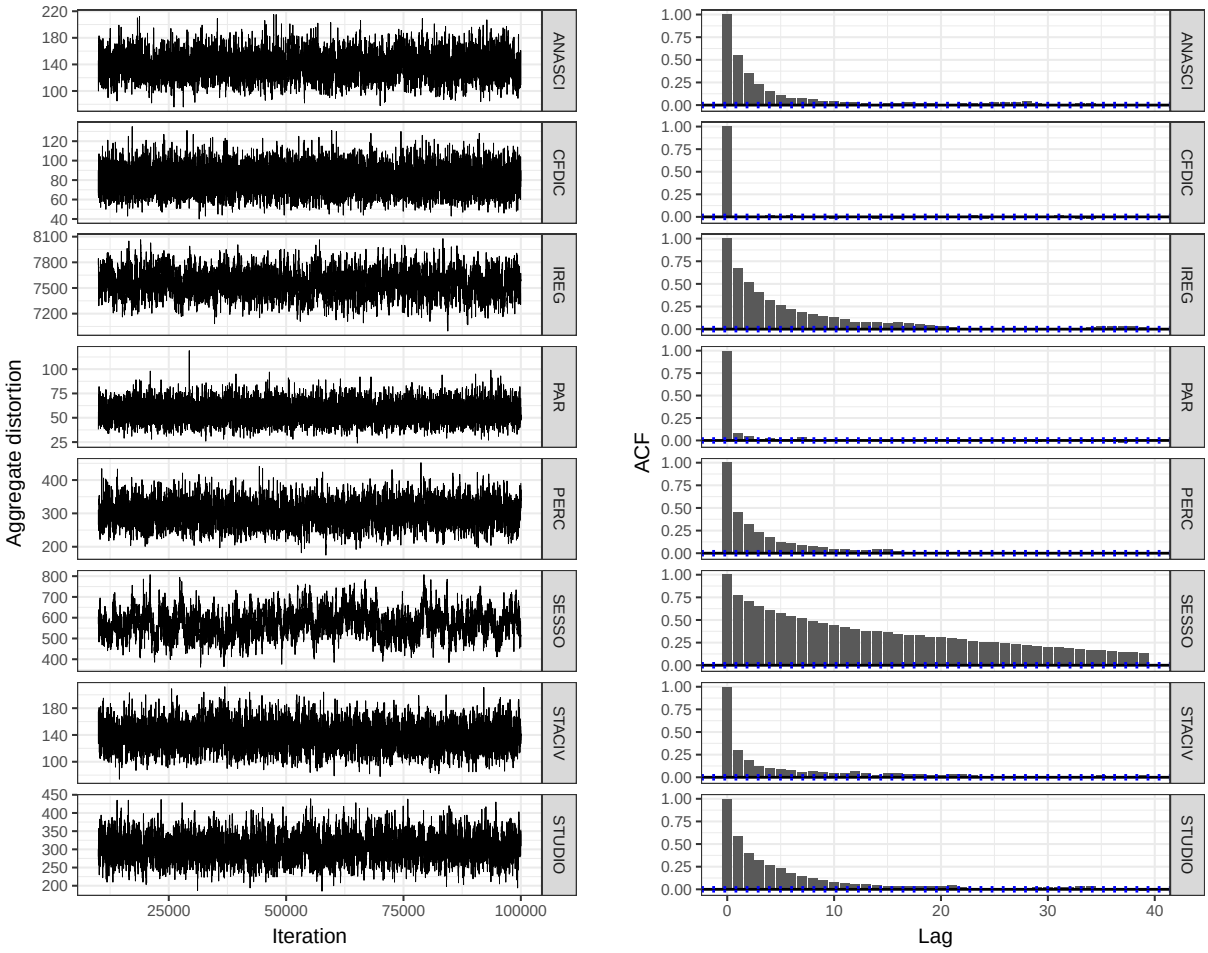


Figure S8: Attribute-level distortion for `SHIW0810`

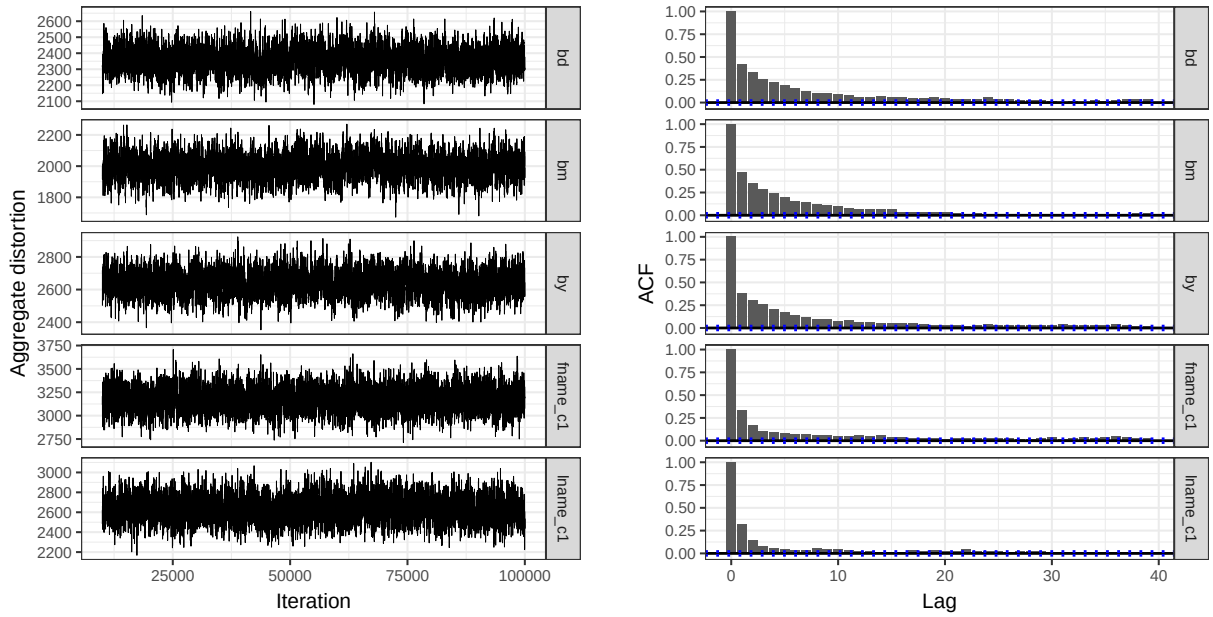


Figure S9: Attribute-level distortion for `RLdata10000`

## L.2 Distribution of record distortion

The following figures relate to the distribution of record distortion for each data set. Specifically, we count the number of records with 0 distorted attributes, 1 distorted attribute, 2 distorted attributes, etc. On the left are the trace plots, which show the record counts for each distortion level (stacked vertically) along the Markov chain. On the right are the corresponding autocorrelation plots.

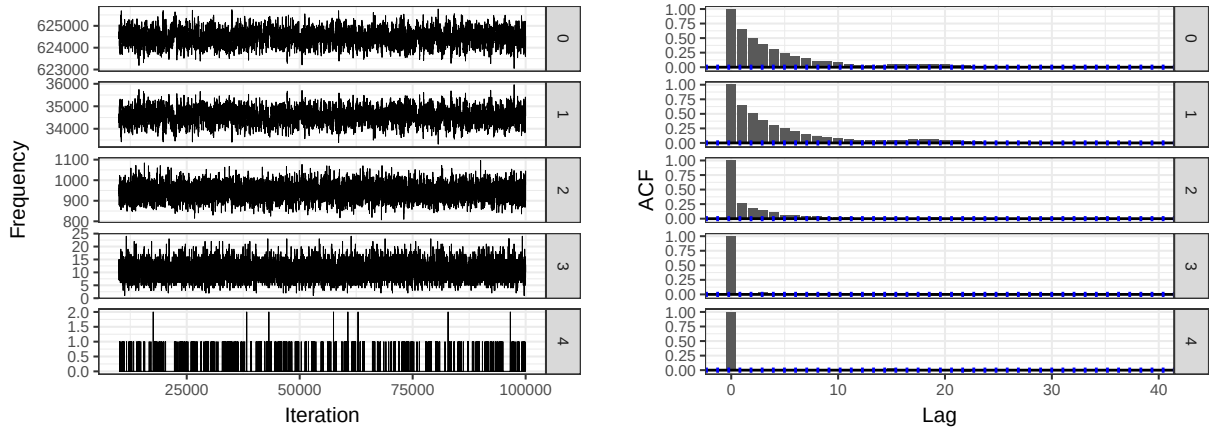


Figure S10: Distribution of record distortion for `ABSEmployee`

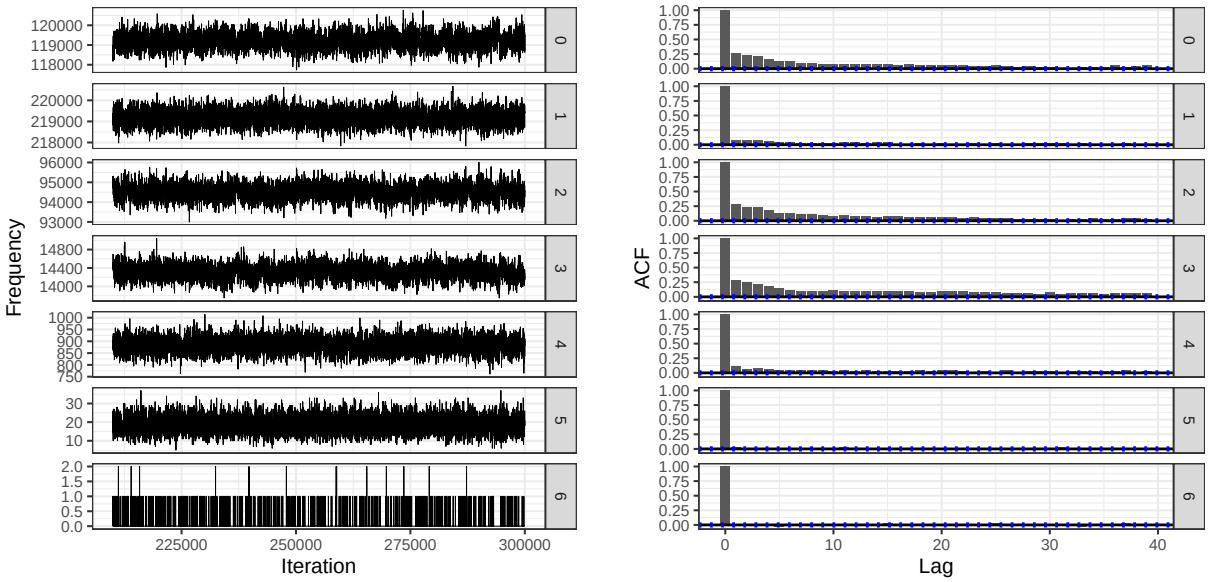


Figure S11: Distribution of record distortion for `NCVR`

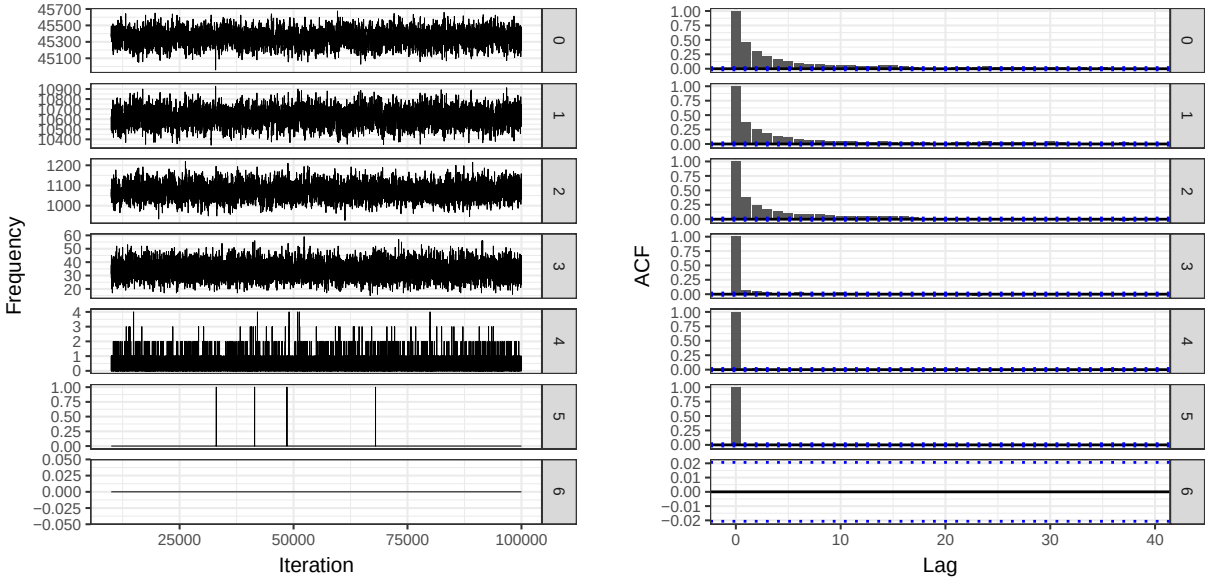


Figure S12: Distribution of record distortion for `NLTCS`

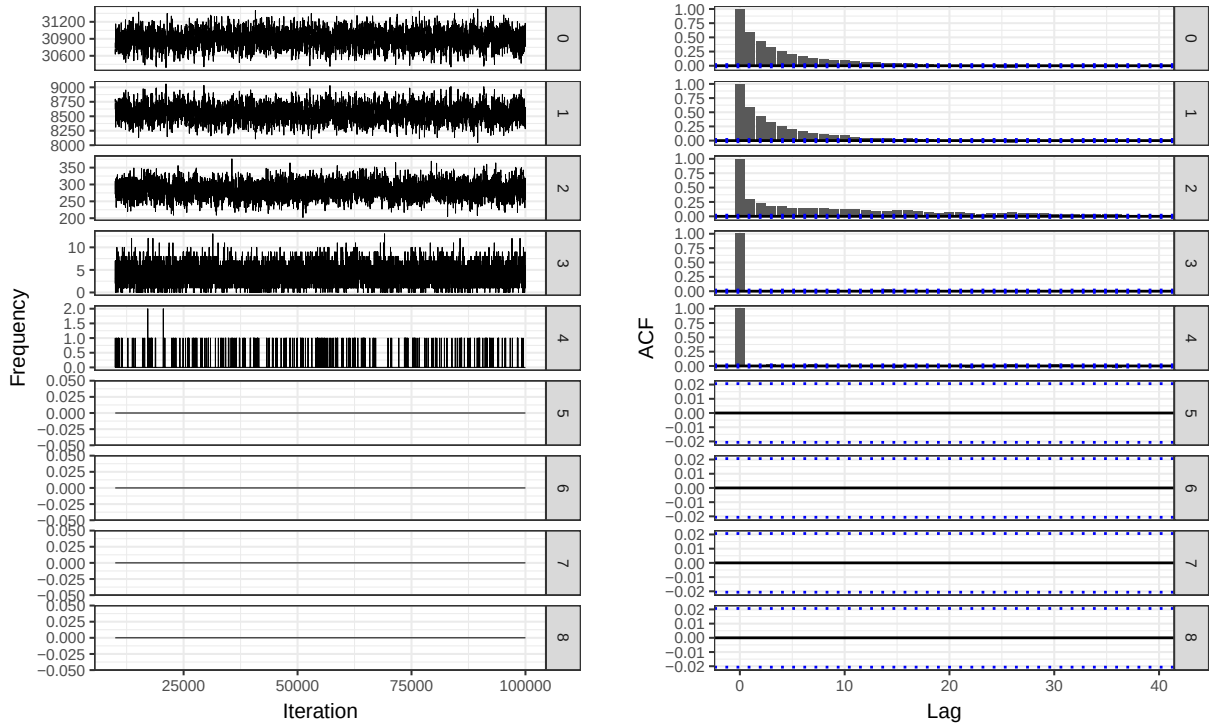


Figure S13: Distribution of record distortion for `SHIW0810`

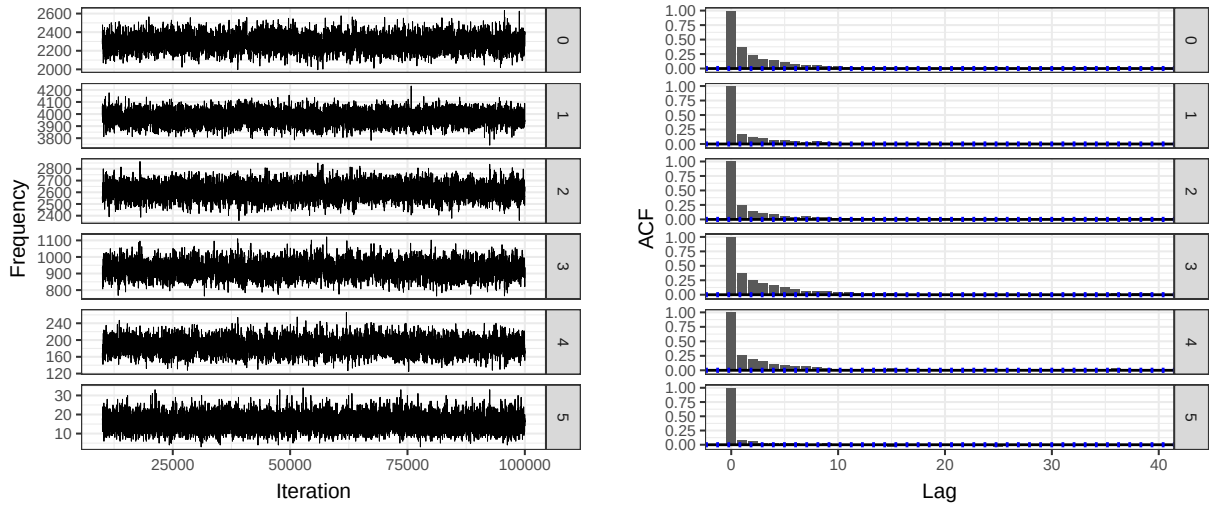


Figure S14: Distribution of record distortion for `RLdata10000`

## L.3 Cluster size distribution

The following figures relate to the distribution of cluster (entity) sizes for each data set. Specifically, we count the number of entities with 0 linked records, 1 linked record, 2 linked records, etc. On the left are the trace plots, which show the counts for each cluster

size (stacked vertically) along the Markov chain. On the right are the corresponding autocorrelation plots.

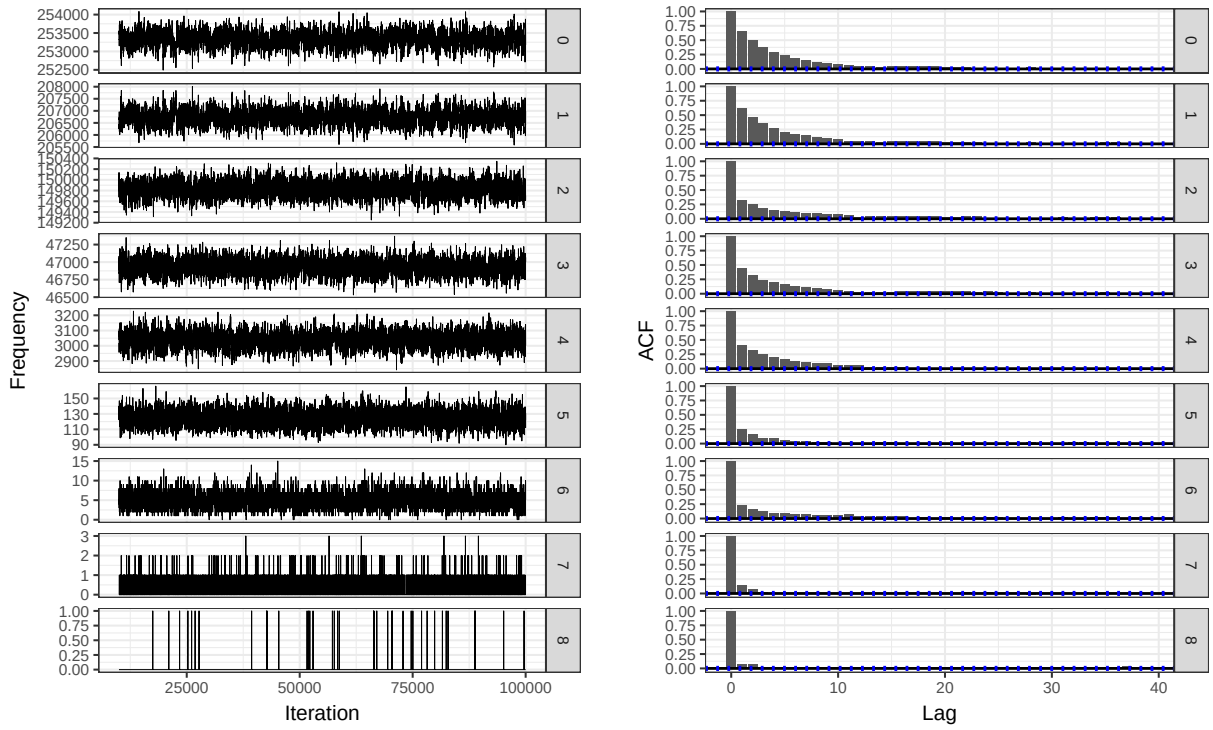


Figure S15: Cluster size distribution for `ABSEmployee`

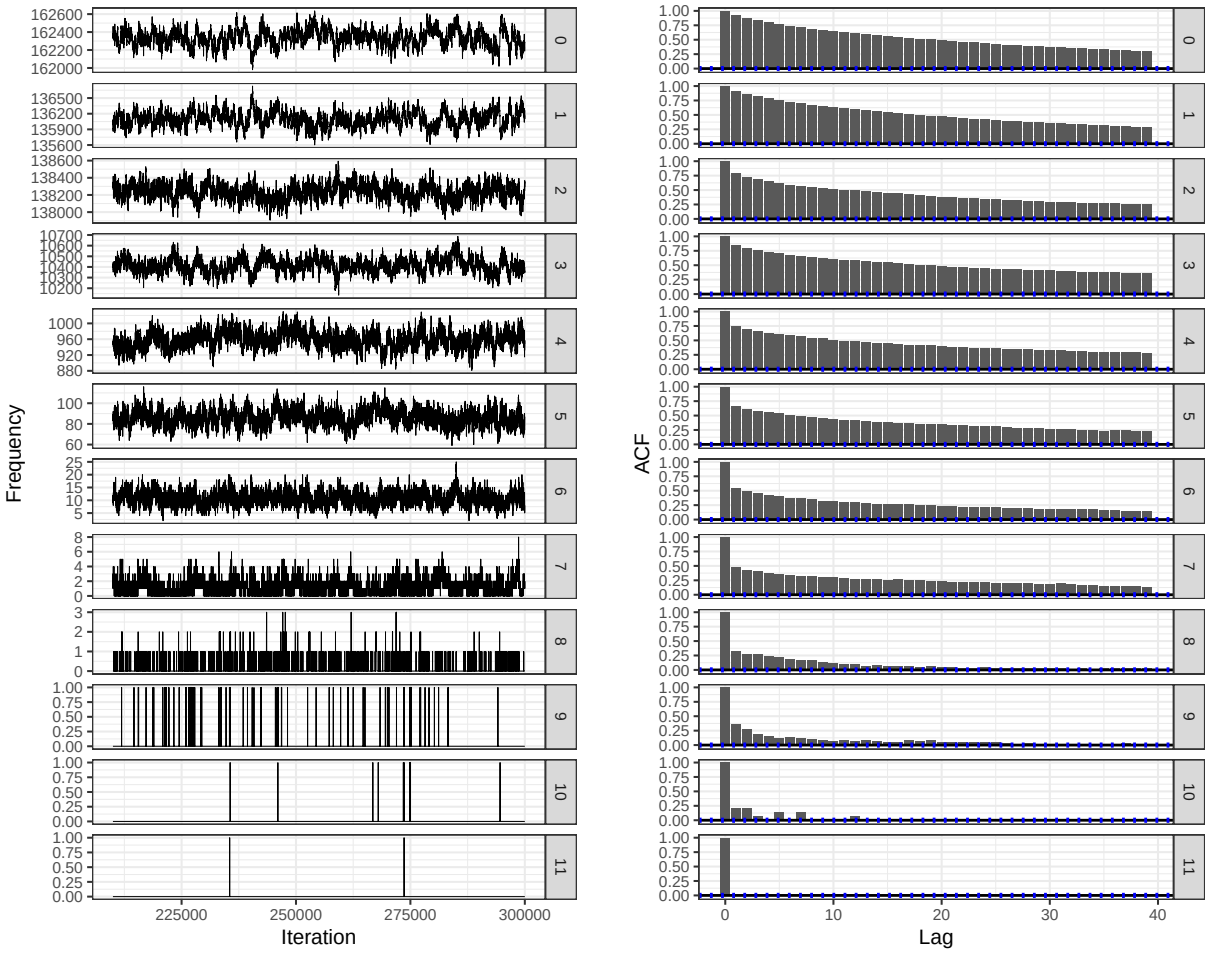


Figure S16: Cluster size distribution for `NCVR`

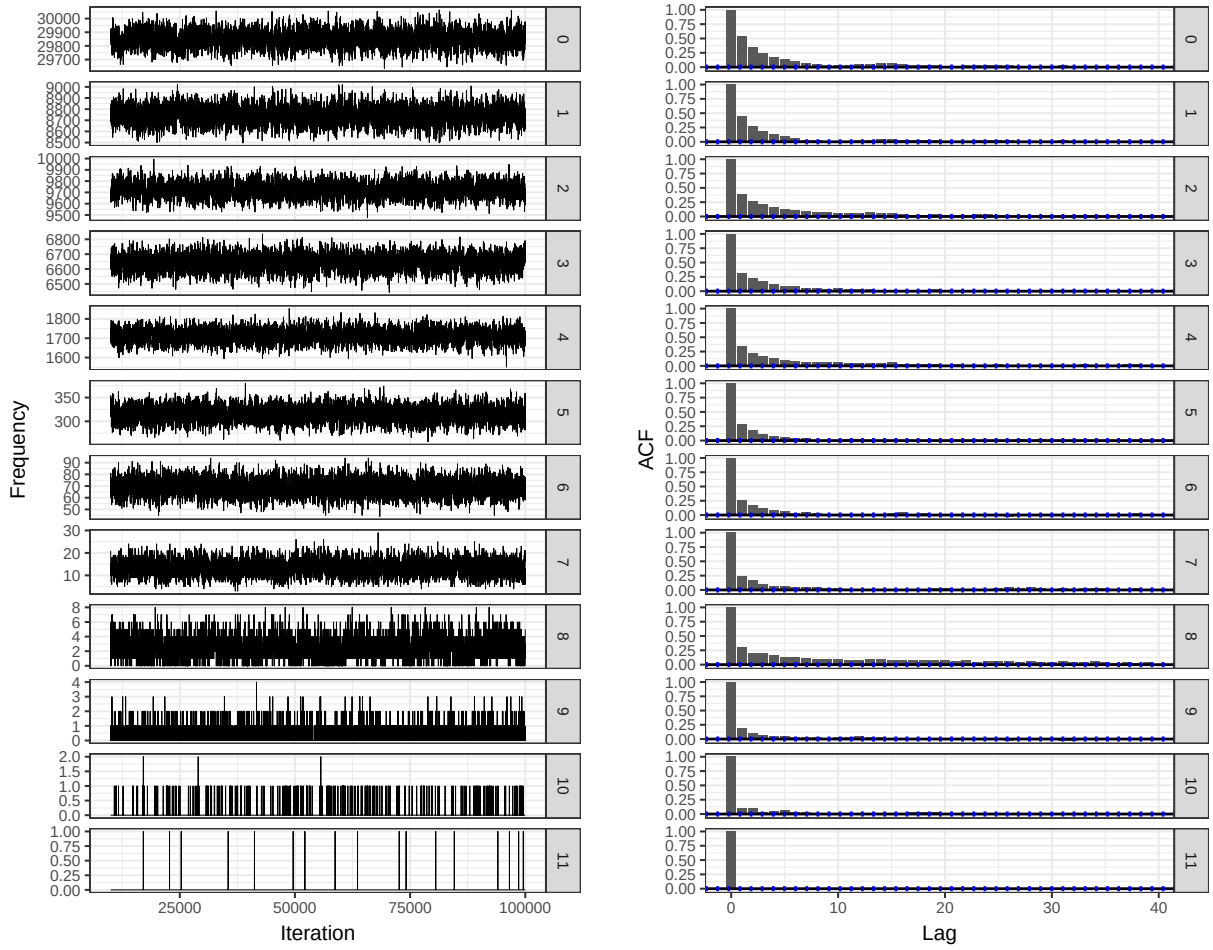


Figure S17: Cluster size distribution for `NLTCS`

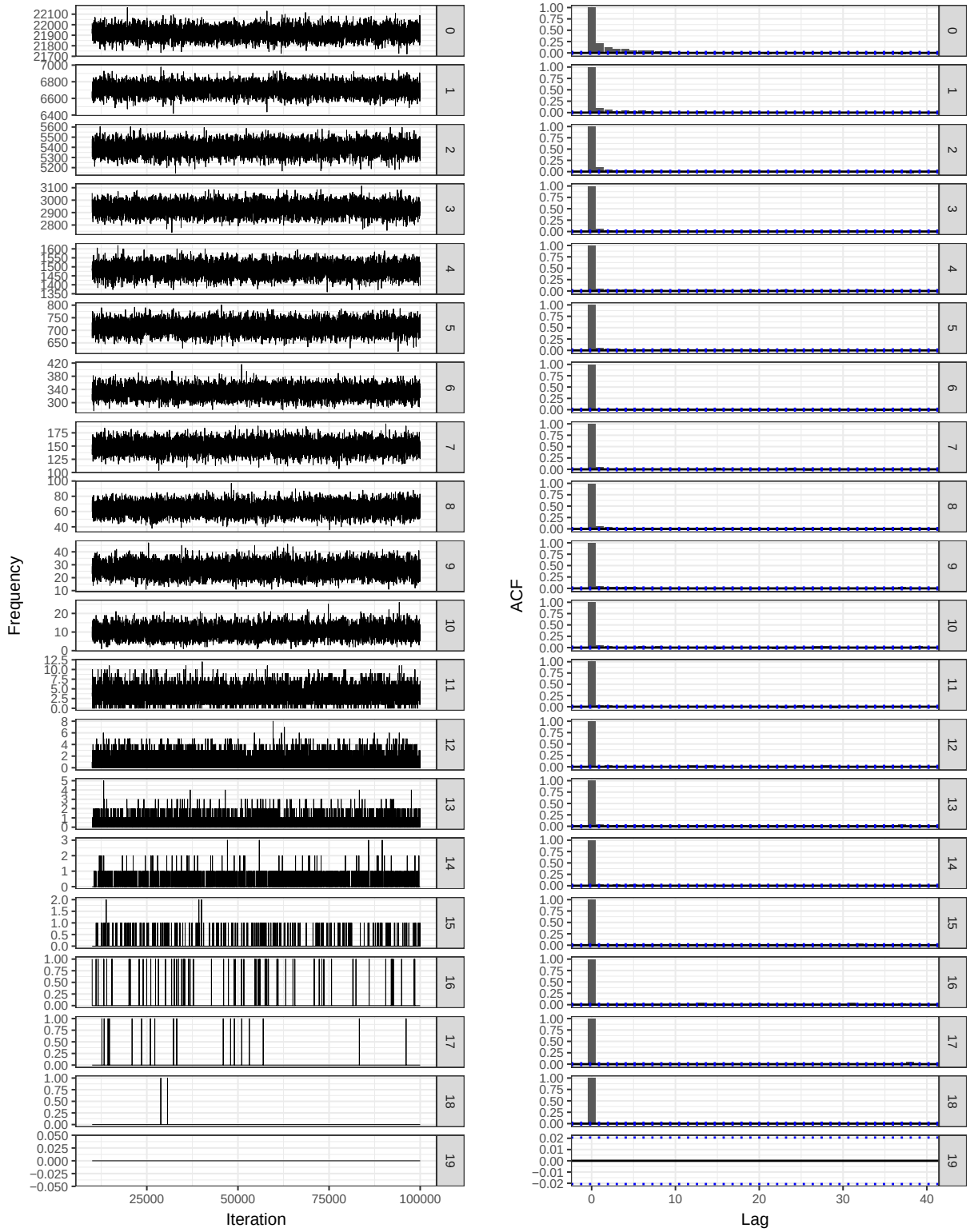


Figure S18: Cluster size distribution for `SHIW0810`

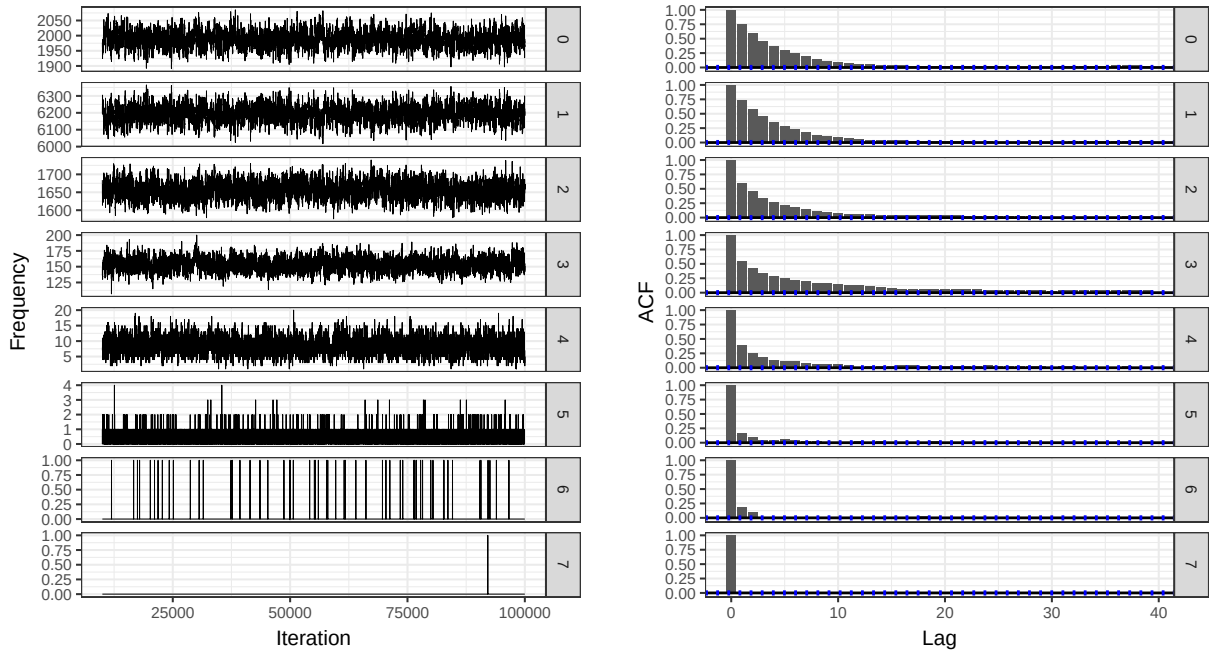


Figure S19: Cluster size distribution for `RLdata10000`

# References

Banca d'Italia. "Bank of Italy – Survey on Household Income and Wealth." http://www.bancaditalia.it/pubblicazioni/indagine-famiglie/index.html (n.d.). Accessed: 2018-03-09.

Christen, P. "Preparation of a real temporal voter data set for record linkage and duplicate detection research." Technical report, Australian National University (2014).

Manton, K. G. "National Long-Term Care Survey: 1982, 1984, 1989, 1994, 1999 and 2004." (2010).

Sariyar, M. and Borg, A. "The RecordLinkage Package: Detecting Errors in Data." *The R Journal*, 2(2):61–67 (2010).

Steorts, R. C., Hall, R., and Fienberg, S. E. "A Bayesian Approach to Graphical Record Linkage and Deduplication." *Journal of the American Statistical Association*, 111(516):1660–1672 (2016).

Vose, M. D. "A linear algorithm for generating random numbers with a given distribution." *IEEE Transactions on Software Engineering*, 17(9):972–975 (1991).

Zanella, G., Betancourt, B., Wallach, H., Miller, J., Zaidi, A., and Steorts, R. C. "Flexible Models for Microclustering with Application to Entity Resolution." In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, 1425–1433. NY, USA: Curran Associates Inc. (2016).