

International Journal of Population Data Science

Journal Website: www.ijpds.org



Towards High Performance Data Curation Statistical Disclosure Control Tooling

Deirdre Lungley¹ and Simon George²

¹UKDS, University of Essex, Colchester, United Kingdom

²Simon George Limited, Wrexham, United Kingdom

Objectives

Surveys are a widely used and important research resource, whose creation and curation involve skilled, labour-intensive tasks. This abstract details an initiative to improve the tooling available to the data community, to control the risk of unintended disclosure, in line with the Anonymisation Decision Making Framework.

Methods

An initial step in assessing dataset disclosivity is the identification of key variables (KVs) (variables which, when combined, can indicate individual units) and the subsequent computation of frequency counts for combinations of these variables. These counts are a prerequisite for achieving k-anonymity, but can also be used in further risk calculations. Their centrality to our processes prompted us to improve the performance of this algorithm. We achieved a significant improvement over the original sdcMicro R package. We do this by decomposing KV values into "bitmasks" (0s and 1s) that are then easily manipulable by native CPU instructions.

Results

In the sdcMicro R package these calculations use the data.table library which, while performant, can be improved upon by our algorithm especially in the common case of the dataset containing missing values.

We tested using the UK Quarterly Labour Force survey, on a Dell XPS 15 9520 laptop. Our implementation makes maximum use of the number of CPU cores and runs the combinations in parallel. For a single combination of 4 KVs we achieve the following¹.

- Time to compute bitmasks 0.408s
- Time to compute frequencies 0.285s
- Total time 0.693s

¹Average over 20 iterations

For all 4-way combinations from the 8 KVs (70 combinations) we achieve the following.

- Time to compute bitmasks 1.489s
- Time to compute frequencies 28.586
- Total time 30.075s

Conclusion

Our chief aim is to contribute code to the community, which allows seamless integration of these performant computations into regular Python applications. Therefore, following publication, this Python wrapped C++ code will be available via GitHub. For broader applicability, the integration of our algorithm into sdcMicro itself could prove useful.

