mtcd is a standalone C++ implementation of the statistical model proposed in “Synthesizing Truncated Count Data for Confidentiality”.

1. The program is writing in ANSI/C++ and can be compiled and ran on all major operating systems such as Linux, Windows and Mac OS. The software is self-contained and does not require any third party library for support.
2. The program is implemented as a command line utility. It does not have a graphical user interface (GUI) and can only be run from a terminal window or from within another program using system calls.

**Software package, Compilation and installation:**

The software is distributed with source code and requires a C++ compiler to generate the binary executable file for targeting platform. It has been tested using gcc for linux and Mac OS and Microsoft Visual C++ for Windows. To compile on linux, simply unzip the package and then type make to get the executable. To compile on Windows, simply add all .cpp and .h files to a new project and compile the project. There is no need to install the program and one can run it by copying it to a working folder or using full path in terminal window.

**Input data:**

The primary input data is a flat text file with each row representing a datapoint. There should be only three columns per row, corresponding to the source county index, destination county index and observed y value. All data should be numeric, with the third column ranging from 1 to 9. An example input file (data.txt) is included in the package.

**Running program:**

The general command line syntax to run the program is

*mtcd –n=num [options] input\_data\_file\_name*

where num is the number of rows in the input data file and the input\_data\_file\_name is the data file to be analyzed. Mtcd currently accepts a few more options. One can run the program without any parameter to get a list of options.

**Options:**

The following options can be used in command line to run the program. All options have long forms and short forms. For example, one can provide a ”Number of observations” option to the command line by setting -n=20 or --nobs=20.

* + -h --help Display this message
  + -n --nobs Number of observations
  + -r --randomseed Random seed
  + -i --iters Number of iterations
  + -b --burnin Number of burnins
  + -s --stride Stride
  + -o --output Output file name
  + -m --nsynthetic Number of synthetic datasets
  + -v --verbose Output debug information

**Output:**

Currently, all outputs are saved into one file under the same directory where the program is running. By default, the output file name is syntheticdata.txt although it can be changed using –o option. The output file is also a spreadsheet like flat text file with each row (except the first row) corresponding to a datapoint from the input data file. The first row is a header row for short column names.

* Column 1:3 (sc,dc,y) the original input data
* The next m columns (sd1,…sdm) are the m synthetic datasets.
* Columns (cl,cu): the 95% confidence interval estimated using all saved models.
* Columns (dt#,Rall#,Runq#): the dt, Rall and Runq risk measure for specific synthetic data set #. Note that Rall and Runq are duplicated in each row, although they are all the same across all the datapoints. (WE MIGHT NEED TO CHANGE IF TOO CONFUSING HERE).

**File list from the zip file:**

* cmdline.h: a helper class for parsing command line options.
* MersenneTwister.h: A Mersenne Twister random number generator.
* model.h: the header file to be used when linking to other programs such as Matlab or R.
* SpecicalFunctions.h, SpecicalFunctions.cpp: a C++ class that implements some functions used by the program. It has more than we need as we only use the gammarand from it. We can remove many functions from it if necessary.
* stfafx.h: a standard header file including all the utilities I need from ANSI C++/
* model.cpp: all functions related to model implementation are implemented here. I did not pack it in a class, so it is much easier to be used as a library to link into Matlab,R or SAS, etc.
* mtcd.cpp: the main function to implement the standalone program.
* data.txt: an example input data file.
* Syntheticdata.txt: an example output datafile.
* makefile: the default makefile for Linux/Mac OS compilation.

Optional output:

When the *–v (verbose)* option is provided, the program will also output two extra files. One is for all saved model parameters and another one is for risk measure probability matrix for individual synthetic data sets.