Replication codes for "Robust Ranking of Happiness Outcomes: A Median Regression Perspective" by Chen, L. Y., Oparina, E., Powdthavee, N., and Srisuma, S. (2022).

This repository contains the codes for estimating semiparametric quantile regression for ordinal outcomes based on mixed integer optimization (MIO). The description of this estimator and its computation details can be found in the paper. The codes include Stata and Matlab files. Implementation of the Matlab function requires the Gurobi solver, which is freely available for academic purposes (you can request the license here). To replicate the results of the paper:

1. Install the Gurobi solver (installation guide here) and link it to Matlab (instruction here)

2. Run the files in the following order:

    - *gss_clean* in Stata. You will need to change the line 9 of the file to ensure that it points to the folder where you extract our files;
    - *bounds.do* in Stata;
    - *empirical_results_median.m* in Matlab;
    - *empirical_results_q25_75.m* in Matlab.

We now describe the codes in this directory:

- **Data cleaning.** The Stata *gss_clean.do* file prepares the GSS dataset obtained from GSS Data Explorer (available here) for the analysis. It also performs necessary adjustment for series break following Stevenson and Wolfers (2008). The code produces *GSS_working.dta* and *GSS_working_even.csv* as outputs.

- **Heteroskedastic ordered probit and logit estimation.** The Stata *bounds.do* file produces the results of homoskedastic and heteroskedastic ordered probit and logit estimates. The results are stored to *ordered_estimates.csv*. It also produces matrices of scaled heteroskedastic ordered probit and logit estimates and their standard errors. These are reported in the paper and used to construct the bounds for the median regression estimation. The results are stored in *logitse.xlsx* and *probitse.xlsx* files, which are used as an input for Matlab estimation codes. The file uses *GSS_working.dta* data file as an input.

- **Median regression estimation** and **Construction of confidence intervals (CI).** The Matlab file *empirical_results_median.m* has two parts. The first one provides point estimates of the parameters of the ordinal discrete choice median regression model. It uses the function defined in the Matlab file *ordered_response_LAD.m*. The code uses *GSS_working_even.csv* data file, and parametric model estimates, *logitse.xlsx* and *probitse.xlsx*, as inputs. It stores the estimates, *bhat*, and other supporting data in the *mio_res_50.mat* file. The second part constructs CI for the parameters of the ordinal discrete choice median regression model. It uses the function defined in the Matlab file *CI_ordinal_LAD.m*. The code stores the estimates, *bhat*, confidence intervals, *CI*, and supporting data in the *mio_res_b_25.mat* file.

- **Estimation and confidence interval construction for other quantiles.** The Matlab file *empirical_results_q25_75.m* performs the same estimation as described in the previous point for the 25th and 75th percentile.

The Matlab codes include options for estimation diagnostics and tuning, that can be useful if you want to use them for a different application. We want to draw particular attention to those related to the computation time. The computational budget for the MIO estimation can be specified in terms of time, T, or the number of nodes, N. Currently the node limit for our MIO computation is set at 10,000 nodes. Setting node limit instead of the time limit allows for replicating the results provided that the user works with the same software environment such as the Gurobi solver version, Matlab version and the operating system. For the results of this paper, we ran the codes using Gurobi 9.5.0 in Matlab 2021b over a PC operated on Windows 10 and equipped with Intel CPU i9-7980XE (2.60 GHz) and 128GB RAM. Note that the results might not be exactly replicated yet would remain similar to those reported in the paper if the codes are run on a different computing environment such as different Gurobi or Matlab versions. MIO computation might take substantial time. For large scale applications, it might be practical to start with setting a time limit, e.g., of one hour, and further explore the performance with different time budgets.

Additionally, some helpful details on the basics of mixed integer programming and its implementation in Gurobi algorithms can be found here and here.

We hope you find these files useful,

Le-Yu Chen, Ekaterina Oparina, Nattavudh Powdthavee, and Sorawoot Srisuma