Replication Package for: A Welfare Analysis of Policies Impacting Climate Change

Robert W. Hahn[*] Nathaniel Hendren[†]
Robert D. Metcalfe[‡] Ben Sprung-Keyser[§]

November 18, 2024

1 Overview

The replication package allows anyone to re-create all figures, tables, and numbers reported in the text. There is also a series of input assumptions that users can change to generate new MVPF estimates. The replication package is structured as follows. Sections 2 and 3 will outline the folder structure for the code and data used for replication. Section 4 will briefly explain how to run the replication package. For users interested in changing the code to reflect alternative specifications, Sections 5 and 6 will discuss the key code files and data files in greater detail.

The code takes significant time (> 1 day) to run the entire replication package, including generating all the estimates with bootstrapped standard errors. As a result, we have provided the processed datasets used in the paper so users can recreate figures, tables, and numbers in-text without regenerating the data. As will be explained in greater detail, users can regenerate the data if desired.

[*] University of Oxford and Carnegie Mellon University
[†] Massachusetts Institute of Technology and Policy Impacts
[‡] Columbia University and NBER
[§] Wharton School, University of Pennsylvania and Policy Impacts

2 Code Structure

- masterfile.do
- ado
  - Stores all of the ado files, which wrapper/macros.do calls in except the cost_curve_masterfile.ado
- bootstrapping
  - Stores the main file that implements bootstrapping.do along with all the files called in bootstrapping.do
- calculations
  - Stores pre-processing files called in macros.do to construct environmental externalities for each policy
- cost_curve
  - Stores the ado file used to calculate cost curve externalities
- data_cleaning
  - Stores pre-processing files used to clean electric vehicle-related data
- figtab
  - Stores all the do files that are needed to create the tables and figures in the paper
- policies
  - The Harmonized subfolder stores all of the policy do-files the main MVPF sample uses
  - The Robustness subfolder stores policy do-files only used for robustness analysis
- publication_bias
  - Stores all the do files that are needed to create the publication bias-corrected version of the estimates
- wrapper
  - Stores the main do files called in masterfile.do, including macros.do, metafile.do, and prepare_causal_estimates.do
  - Stores the do files used to create all the main and appendix figures and tables
  - Stores robustness.do, which creates all the robustness numbers reported in the text

3 Data Structure

This list includes all the subfolders of the data folder.

- 0_log
  - After each run of the metafile, a new log file will be created and stored here
- 1_assumptions
  - Stores all assumptions used in the externality calculations.
- 2a_causal_estimates_papers
  - This file stores the causal estimates used in each policy analyzed. The name of the spreadsheet corresponds to the name of the policy do-file.
- 2b_causal_estimates_draws
  - This folder stores the bootstrapped causal estimates that are needed when calculating the bootstrapped standard errors. Each metafile run will produce a new subfolder named using the run's timestamp and the run's name. If the reps argument passed into the metafile is set to 0, the rest of the code will only use the point estimate of the causal estimate for each policy.
- 3_bootstrap_draws
  - Similar to 2b_causal_estimates_draws, each metafile run will produce a new subfolder that is named using the run's timestamp and the run's name. Instead of storing the bootstrapped causal estimates, this folder will store the bootstrapped MVPF components.
- 4_results
  - Similar to the folders above, each metafile run will produce a new subfolder in 4_results that is named using the run's timestamp and the run's name. This folder will have the main compiled results from the run. In other words, this file will store the MVPF along with the components used to construct the MVPF for each policy included in the run.
- 5_graphs
  - This folder stores the main and appendix figures reported in the paper. It also stores some processed data used to create the figures.
- 6_tables

  - This folder stores the main and appendix tables reported in the paper. It also stores some processed data that the code uses to create the tables.

- 7_publication_bias
  - Stores some data used to create the publication bias results.

4 How to Run

First, download Mathematica and MATLAB to your computer if you have not already done so. To replicate the results, open masterfile.do in the code folder. You should first change the working directory on line XX to the folder's location on your computer. Next, you should decide if you want to simply recreate the figures, tables, and numbers reported in the text or if you want to regenerate the underlying datasets. If you only want to replicate the figures, tables, and robustness numbers, you should set the rerun_data global on line XX to "no" and the bootstraps global on line XX to "no". Once you do this, you can run the masterfile. The code exports the figures to the graphs subfolder and the tables to the tables subfolder. The code exports the robustness numbers reported in the text to an Excel called XX.

Alternatively, you can also regenerate the data used for the analysis. You should set the rerun_data global mentioned above to "yes" to do this. If you would also like to regenerate the bootstraps, you can set the bootstraps global to "yes". Lines XX - XX determine which specifications you want for each run. If you keep the current code as is, the masterfile will run all seven specifications used in the paper. However, you also have the option to alter these specifications. You can determine the mode (current or baseline)[1], the social cost of carbon (76, 193, or 337), whether to include learning by doing (yes or no), whether to include profits (yes or no), whether to include savings (yes or no), which policies to run (all_programs or a specific list of policies), and the name of the run. The second to last positional argument, reps, should always be set to zero as bootstrapping occurs in the next section of the code. After deciding on the set of runs, you can run masterfile.do. If you are regenerating all the data and the bootstraps, you can expect this code to take over 24 hours to run. Once finished, the resulting dataset will be populated in the 4_results folder with the timestamp and name of the run. The code stores the bootstrapped MVPFs and category averages in 5_graphs/figures_data with the v2 file name. Note that if you change the specifications of the runs, the figures, tables, and robustness numbers will not reflect the changes. The code creates them using the standard specifications to match the paper's results.

If you are only interested in replicating the results in the paper, there is no need to read Section 5. However, if you are interested in altering/understanding the code base, Section 5 briefly explains the important code files.

[1] Current references the 2020 specification, and baseline references the in-context specification. Most of the results reported in the paper use the current specification.

5 Explanation of Important Code Files

Macros.do This file, stored in the wrapper subfolder, calculates all of the externalities used for each policy's MVPF. Therefore, any change in specifications requires the user to re-run macros.do. If you are running through masterfile.do, it will automatically call macros.do. Macros.do creates the electricity, natural gas, and vehicle-related externalities.

Policy do files In the code/policies/harmonized subfolder, there is a separate do file for each policy used in the analysis. For a mapping of the policy do-file name to the policy name reported in Table One, see files/policy_details_v3.xlsx.

You cannot run each policy do-file alone. Instead, the masterfile must call it. To run an individual policy, a user can change the all_programs positional argument in the masterfile to the name of the specific policy. Each policy loads in the causal effect estimates from the relevant paper(s). These may be the adjusted causal estimates when run externally for bootstrapping or publication bias. Next, the policy do-files calculate each policy's WTP and net government cost. The do-files call on globals, datasets, and ado files introduced in macros.do to do this. Finally, the last section computes the MVPF as the ratio of the WTP and cost and stores all the relevant MVPF components as globals. Many do-files also store components you can use to create policy waterfall graphs if desired.

Prepare Causal Estimates Before estimating MVPFs and bootstrapping them, we prepare bootstrap draws of the causal estimates that go into each MVPF calculation. The file prepare_causal_estimates.do does this by looping over all the policies in our sample. For each policy, it finds the corresponding causal estimates file in causal_estimates/uncorrected, which details point estimates and standard errors. (In cases where standard errors are unavailable, we use t-statistics or p-values to back them out. Where the paper only reports significance levels, we draw uniformly over the range of possible associated p-values. We record the standard error as 0 in cases where the authors do not report anything. We do not use policies with missing standard errors when calculating standard errors.)

The file details the assumed correlational structure between estimates in the column ”corr_direction.” This variable defines blocks in the correlation matrix between estimates: we indicate blocks by different base numbers, and the sign determines the correlation direction. E.g., if we have four variables with corr_directions 1, -1, 2, 2 respectively, then 1 and 2 are perfectly negatively correlated, 3 and 4 are perfectly positively correlated, and 1 and 2 are uncorrelated with 3 and 4. We set these correlations to maximize the width of our confidence intervals where estimates are from the same sample. Given point estimates, standard errors, and a correlation matrix, we draw from the implied joint normal distribution 1000 times, generating the 1000 sets of estimates corresponding to our 1000 bootstrap estimates of the MVPF.

Bootstrapping To get confidence intervals for the MVPF estimates, we bootstrap standard errors. This process occurs in bootstrapping.do. Since learning-by-doing takes significantly more run time, we treat policies with and without learning-by-doing separately. For policies without learning-by-doing, we run the metafile with 1000 replications. The causal estimates are created in prepare_causal_estimates and fed into bootstrap_wrapper. For policies with learning-by-doing, we loop through a series of causal estimates to understand the functional form between the causal estimate and the resulting MVPF. Once we have estimated this functional form, we use it to create the bootstrapped MVPFs without running the metafile on 1000 replications. This entire process occurs in bootstrapping.do.

Publication Bias To estimate MVPFs corrected for publication bias, we run pub_bias_wrapper, which calls four nested do-files. prep_and_run_matlab estimates the level of publication bias in our sample of environmental policies. cdf_plot generates the implied CDFs from our method of estimating publication bias and from that of Andrews and Kasy (2019), both compared to the empirical CDF of the (absolute value of the) t-stats in our data. heuristic_graphs generates the "funnel plot" of the standard errors for the studies in our sample against the corresponding point estimates. Lastly, replace_w_corrected corrects the point estimates of the papers in our sample and edits Excel sheets that the code uses in MVPF calculations.

Robustness This file, stored in the wrapper subfolder, calculates the numbers reported in the text that we do not directly take from a table or figure. These numbers are outputted to the 4_results subfolder and saved as robustness.xlsx.

Figure & Table Creation We store the wrapper files that create all the tables and figures in the wrapper subfolder as tables.do, figures.do, appendix_tables.do, and appendix_figures.do. Each do-file has nested files in the figtab folder, which construct the individual figures and tables. All the tables and figures are stored in 5_graphs and 6_tables, respectively.

6 Explanation of Important Data Files

Policy Assumptions Masterfile The assumptions that we use to calculate the environmental and fiscal externalities are primarily pulled from the 1_assumptions/policy_category_assumptions_MASTER.xlsx spreadsheet. We list the sources for the data pulled from the spreadsheet on the first tab.

Program Specific Causal Estimates The causal estimates used for each policy are stored in 2a_causal_estimates_papers/uncorrected_vJK. The names of the spreadsheets correspond to the name of each policy do-file. On the raw_data tab of each spreadsheet, we list where we are pulling the estimate in the paper, the link to the paper PDF, and the date we accessed it. We also show any transformations to the causal estimate before it enters the policy do-file.

References

Isaiah Andrews and Maximilian Kasy, “Identification of and Correction for Publication Bias,” *American Economic Review*, vol. 109, no. 8, pp. 2766–2794, August 2019.