

# A More Credible Approach to Parallel Trends: Replication Files

Ashesh Rambachan and Jonathan Roth

12/2022

This provides a description of the replication files for “A More Credible Approach to Parallel Trends” by Ashesh Rambachan and Jonathan Roth.

## Software requirements

To run these files, please install the R package `HonestDiD`, which is available for download on Github. The vignette available on Github provides instructions on how to install the latest version of the package and a detailed description of how to use the package.

In order to replicate the results in the paper, please use the following code to install the appropriate version of the `HonestDiD` package.

```
## Installation

# Install remotes package if not installed
install.packages("remotes")

# Turn off warning-error-conversion, because the tiniest warning stops installation
Sys.setenv(R_REMOTES_NO_ERRORS_FROM_WARNINGS = "true")

# install from github using correct version.
remotes::install_github("asheshrambachan/HonestDiD", ref = "v1.0.0")
```

We also recommend that the user installs the R package [here](#). This will enable easy file referencing. See the documentation for details.

The R package `HonestDiD` relies on several other R packages that will be installed automatically in the installation of `HonestDiD` if they are not already installed. We list these additional R packages, and provide code on how to install them here for completeness.

```
## Installation of additional dependencies

install.packages("tidyverse") # installs R package, tidyverse for data manipulation
install.packages("CVXR") # installs R package, CVXR for convex optimization

# installs R package, doParallel for easy parallelization
install.packages("doParallel")

install.packages("foreach") # installs R package, foreach for loops
install.packages("latex2exp") # installs R package, latex2exp for use of latex in figures
install.packages("lpSolveAPI") # installs R package, lpSolveAPI for linear programming
install.packages("Matrix") # installs R package, Matrix for matrix calculations
install.packages("pracma") # installs R package, pracma for linear algebra calculations
install.packages("purrr") # installs R package, purrr for vectorization
install.packages("ROI") # installs R package, ROI for linear/nonlinear optimization
```

```
# installs R package, TruncatedNormal for truncated normal distribution functions
install.packages("TruncatedNormal")
```

## File Structure

The replication files are organized into the following main directories:

- **/Code:** contains all R scripts and bash scripts used to construct the empirical analyses and simulation study.
- **/Figures:** contains all final figures in the paper.
- **/Inputs:** contains all input files that are used to construct the empirical analyses and simulation study.
- **/Outputs:** contains all outputted dataframes that are produced in the empirical analyses and simulation study.
- **/Temp:** contains all intermediate dataframes that are produced in the empirical analyses and simulation study.
- **/Tables:** contains tables in the paper.

In the next sections, we discuss how to replicate each component of the paper using the replication files.

## Lovenheim & Willen (2019) analysis

The code needed to replicate the analysis of Lovenheim & Willen (2019a) are contained in the directory `/Code/LovenheimWillen-Scripts`. To replicate the analysis, the user only needs to run the R script, `/Code/LovenheimWillen-Scripts/LovenheimWillen-Analysis.R`. This reconstructs the employment event studies using the raw data, constructs the original event study plots, constructs robust confidence intervals and constructs the sensitivity plots contained in Rambachan & Roth (2022). Run the R command:

```
source(here("Code/LovenheimWillen-Scripts/LovenheimWillen-Analysis.R"))
```

The results are stored into the directory `Figures/`. The results take less than 15 minutes to compute.

## Benzarti & Carloni (2019) analysis

The code needed to replicate the analysis of Benazarti & Carloni (2019a) are contained in the directory `/Code/BenzartiCarloni-Scripts`. To replicate the analysis, the user only needs to run the R script, `/Code/BenzartiCarloni-Scripts/BenzartiCarloni-Analysis.R`. This script will reconstruct the profit event studies using the estimated coefficients and covariance matrix in Benazarti & Carloni (2019a), construct the original event study plot, construct robust confidence intervals and construct the sensitivity plots contained in Rambachan & Roth (2022). The authors were kind enough to share the estimated coefficients and covariance matrix with us since the original data are confidential. Run the R command:

```
source(here("Code/BenzartiCarloni-Scripts/BenzartiCarloni-Analysis.R"))
```

The results are stored into the directory `Figures/`. The results take less than 15 minutes to compute.

## Simulation study: normal simulations

To replicate the figures in the paper using the stored simulations results (which we computed on a high-performance computing cluster), run the following R commands. (We describe how to re-run all simulation output below).

```
# Delta^{SD}
source(here(paste0("Code/simulationStudy/AnalyzeResults_ParallelTrends/", "simulationStudy_analyze_Delta^{SD}")))
source(here(paste0("Code/simulationStudy/AnalyzeResults_ParallelTrends/", "simulationStudy_analyze_Delta^{SD}")))

# Delta^{SDPB}
```

```

source(here(paste0("Code/simulationStudy/AnalyzeResults_ParallelTrends/", "simulationStudy_analyze_DeltaSD_FirstPostPeriod_ParallelTrends.R")))
source(here(paste0("Code/simulationStudy/AnalyzeResults_ParallelTrends/", "simulationStudy_analyze_DeltaSD_Avg_ParallelTrends.R")))

# Delta~{RM}
source(here(paste0("Code/simulationStudy/AnalyzeResults_VaryPulse/", "simulationStudy_analyze_DeltaRM_FirstPostPeriod_varyPulse.R")))
source(here(paste0("Code/simulationStudy/AnalyzeResults_VaryPulse/", "simulationStudy_analyze_DeltaRM_Avg_varyPulse.R")))

# Delta~{SDRM}
source(here(paste0("Code/simulationStudy/AnalyzeResults_VaryPulse/", "simulationStudy_analyze_DeltaSDRM_FirstPostPeriod_varyPulse.R")))
source(here(paste0("Code/simulationStudy/AnalyzeResults_VaryPulse/", "simulationStudy_analyze_DeltaSDRM_Avg_varyPulse.R")))

```

The figures are saved into the directory **Figures/**.

We now describe how to conduct the simulations for a large number of simulation draws as in the paper. The simulation study code is structured to be run on a high-performance computing cluster environment that uses a slurm scheduler. To construct the simulation results, first run the R commands to redraw the normal simulations:

```

source(here("Code/simulationStudy/simulationStudy_drawEventStudyCoefficients_parallelTrends.R"))
source(here("Code/simulationStudy/simulationStudy_drawEventStudyCoefficients_varyPulseDesign.R"))

```

To construct the simulation results on excess length for the first post-period estimand over  $\Delta^{SD}(M)$ ,  $\Delta^{SDPB}(M)$ ,  $\Delta^{RM}(\bar{M})$ , and  $\Delta^{SDRM}(\bar{M})$ , run the bash commands:

```

sbatch Code/batchScripts/runSimStudy_DeltaSD_FirstPeriod_ParallelTrends.sh # Delta~{SD}
sbatch Code/batchScripts/runSimStudy_DeltaSDPB_FirstPeriod_ParallelTrends.sh # Delta~{SDPB}
sbatch Code/batchScripts/runSimStudy_DeltaSDRM_FirstPeriod_VaryPulse.sh # Delta~{SDRM}
sbatch Code/batchScripts/runSimStudy_DeltaRM_FirstPeriod_VaryPulse.sh # Delta~{RM}

```

To construct the simulation results on excess length for the average post-period estimand over  $\Delta^{SD}(M)$ ,  $\Delta^{SDPB}(M)$ ,  $\Delta^{RM}(\bar{M})$ , and  $\Delta^{SDRM}(\bar{M})$ , run the bash commands:

```

sbatch Code/batchScripts/runSimStudy_DeltaSD_Avg_ParallelTrends.sh # Delta~{SD}
sbatch Code/batchScripts/runSimStudy_DeltaSDPB_Avg_ParallelTrends.sh # Delta~{SDPB}
sbatch Code/batchScripts/runSimStudy_DeltaSDRM_Avg_VaryPulse.sh # Delta~{SDRM}
sbatch Code/batchScripts/runSimStudy_DeltaRM_Avg_VaryPulse.sh # Delta~{RM}

```

These batch scripts execute a series of slurm commands that submit jobs to re-construct the normal simulations. Note that these scripts may need to be modified depending on the parameters of the computing environment (i.e., the memory, cores, and run time may need to be modified). The running time of these simulations depends on your computational resources. Each bash script takes about 1-2 weeks to run. Once these slurm commands have been completed, execute the following commands in R to combine the simulation results and construct the figures:

```

# Delta~{SD}
source(here("Code/simulationStudy/simulationStudy_analyze_DeltaSD_FirstPostPeriod_ParallelTrends.R"))
source(here("Code/simulationStudy/simulationStudy_analyze_DeltaSD_Avg_ParallelTrends.R"))

# Delta~{SDPB}
source(here("Code/simulationStudy/simulationStudy_analyze_DeltaSDPB_FirstPostPeriod_ParallelTrends.R"))
source(here("Code/simulationStudy/simulationStudy_analyze_DeltaSDPB_Avg_ParallelTrends.R"))

# Delta~{RM}
source(here("Code/simulationStudy/simulationStudy_analyze_DeltaRM_FirstPostPeriod_varyPulse.R"))
source(here("Code/simulationStudy/simulationStudy_analyze_DeltaRM_Avg_varyPulse.R"))

# Delta~{SDRM}

```

```
source(here("Code/simulationStudy/simulationStudy_analyze_DeltaSDRM_FirstPostPeriod_varyPulse.R"))
source(here("Code/simulationStudy/simulationStudy_analyze_DeltaSDRM_Avg_varyPulse.R"))
```

## Simulation study: bootstrap simulations

To replicate the figures in the paper using the stored simulations results (which we computed on a high-performance computing cluster), run the following R commands. (We describe how to re-run all simulation output below).

```
# Delta^{SD} and Delta^{SDPB}
source(here("Code/Bootstrap_ExcessLengthSims/bootstrapNormalComparisonPlots-DeltaSD_DeltaSDPB.R"))
source(here("Code/Bootstrap_ExcessLengthSims/bootstrapSizeControlTables-DeltaSD_DeltaSDPB.R"))

# Delta^{RM} and Delta^{SDRM}
source(here("Code/Bootstrap_ExcessLengthSims/bootstrapNormalComparisonPlots-DeltaRM_DeltaSDRM.R"))
source(here("Code/Bootstrap_ExcessLengthSims/bootstrapSizeControlTables-DeltaRM_DeltaSDRM.R"))
```

The figures are saved into the directory **Figures/**. The tables are saved into the directory **Tables/**

To reconstruct the simulation results, first run the following R command to reproduce the bootstrap draws:

```
source(here("Code/Bootstrap_ExcessLengthSims/BaileyGoodmanBacon-constructBootstrapSamples.R"))
```

The simulation study code is structured to be run on a high-performance computing cluster environment that uses a slurm scheduler. To construct the bootstrap simulation results over  $\Delta^{SD}(M)$ ,  $\Delta^{SDPB}(M)$ ,  $\Delta^{RM}(\bar{M})$ , and  $\Delta^{SDRM}(\bar{M})$ , run the bash command:

```
sbatch Code/batchScripts/runBootstrapSimulations.sh
```

This batch scripts execute a series of slurm commands to re-construct the bootstrap simulations. Note that these scripts may need to be modified depending on the computing environment (i.e., the memory, cores, and run time may need to be modified). The running time of these simulations depends on your computational resources. Each bash script takes about 1-2 weeks to run. Once these slurm commands have been completed, execute the following commands in R to construct the figures and tables:

```
# Delta^{SD} and Delta^{SDPB}
source(here("Code/Bootstrap_ExcessLengthSims/bootstrapNormalComparisonPlots-DeltaSD_DeltaSDPB.R"))
source(here("Code/Bootstrap_ExcessLengthSims/bootstrapSizeControlTables-DeltaSD_DeltaSDPB.R"))

# Delta^{RM} and Delta^{SDRM}
source(here("Code/Bootstrap_ExcessLengthSims/bootstrapNormalComparisonPlots-DeltaRM_DeltaSDRM.R"))
source(here("Code/Bootstrap_ExcessLengthSims/bootstrapSizeControlTables-DeltaRM_DeltaSDRM.R"))
```

## Data availability

The replication files includes the raw data from Lovenheim & Willen (2019a) in the directory **Inputs/LovenheimWillen** of the replication files (Lovenheim & Willen 2019b).

The replication files include the estimated coefficients and covariance matrix from Benzarti & Carloni (2019a), which the authors were kind enough to share. These are provided in the directory **Inputs/BenzartiCarloni** of the replication files. The original data used in Benzarti & Carloni (2019b) are confidential, and we refer the user to the replication files in Benzarti & Carloni (2019b) for further details.

## Configuration

These replication files were last run on a 2021 MacBook Pro (14") with an Apple M1 Max chip, 10 CPUs, 32 GB of RAM, and 1 TB SSD.

## References

- Benzarti, Youssef and Dorian Carloni. “Who Really Benefits from Consumption Tax Cuts? Evidence from a Large VAT Reform in France.” *American Economic Journal: Economic Policy*, 2019(a), 11 (1), 38-63.
- Benzarti, Youssef, and Carloni, Dorian. “Replication data for: Who Really Benefits from Consumption Tax Cuts? Evidence from a Large VAT Reform in France,” 2019(b), American Economic Association [publisher], Inter-university Consortium for Political and Social Research [distributor]. <https://doi.org/10.3886/E114723V1>.
- Lovenheim, Michael F. and Alexander Willen. “The Long-Run Effects of Teacher Collective Bargaining.” *American Economic Journal: Economic Policy*, 2019, 11 (3), 292-324.
- Lovenheim, Michael F., and Willen, Alexander. “Replication data for: The Long-Run Effects of Teacher Collective Bargaining,” 2019(b), American Economic Association [publisher], Inter-university Consortium for Political and Social Research [distributor]. <https://doi.org/10.3886/E116525V1>.
- Rambachan, Ashesh and Jonathan Roth. “A More Credible Approach to Parallel Trends.” 2022.