Replication Package

Price Floors and Employer Preferences: Evidence from a Minimum Wage Experiment

by **John Horton**

1. Overview

This document describes the organization of code, data, and software associated with "Price Floors and Employer Preferences: Evidence from a Minimum Wage Experiment" (Horton, 2025a). The replication package for this project is available at the Github repository johnjosephhorton/minimum_wage (Horton, 2025b) as well as the openICPSR repository 208551V1 (Horton, 2025c).

This project is largely self-documenting in the sense that the key recipes for building the project are captured in code. The key pieces for replicating this project are:

1. A Makefile that gives recipes for how each figure, table, and called-out number is constructed in the paper.

2. A Dockerfile that describes the system set-up needed with respect to computing resources and packages.

A comprehensive list of files, programs, and software dependecies is provided in the current document as well for the reader's convenience.

2. Folder Structure

The main directory contains three folders:

```
-analysis
-codebooks
-writeup
```

The `analysis` folder contains 23 R scripts that generate all outputs used in the paper. The complete list of programs as well as software requirements and dependencies is given in section 4.1.

The `codebooks` folder contains 10 Markdown files that give the overview of the contents of each dataset used in the analysis (one Markdown file for each dataset). The complete list of datasets used in this project is given in section 4.2.

The `writeup` folder contains the main writeup of the paper (`minimum_wage.tex`). The figures, tables, and parameter files generated during replication will be stored in `writeup/plots`, `writeup/tables`, and `writeup/parameters`, respectively. In the repositories hosting the replication package, these locations are empty by design. When the project is built, these folders are populated with the necessary files.

When the project is building (more on this in section 3), a `data` folder is automatically created in the main directory and populated with the relevant datasets. (There is no need to create an empty `data` folder manually.)

3. Replication

The build process for this project is managed using "make" – a tool for automating workflows. A replicator will obtain a `.env` file from the author and can reproduce the entire writeup of the paper via a push-button approach using one of three available options brought in section 3.2.

3.1. Makefile Logic

The Makefile in the `writeup` folder specifies the complete list of file dependencies for replication of the project. A comprehensive list of all dependencies is provided in section 4 of the current readme as well. Below I provide a short tutorial on `make`. Familiar readers may skip the tutorial and go to section 3.2 directly.

3.1.1. A Short Make Tutorial

A Makefile lists recipes for how a particular output used in the paper is constructed. The Makefile entry for the `first_stage.pdf` in the `plots` folder looks like this:

```
plots/first_stage.pdf: ../analysis/first_stage.R  data/df_mw_first.csv
 cd ../analysis && Rscript first_stage.R
```

Note that the target, or output is `plots/first_stage.pdf`. After the colon is what this output depends on, code-wise, which is an R file in the `analysis` folder called `first_stage.R`. The tabbed-in-line below is the recipe for how the target is constructed. In this case, it is just by running `Rscript` on `first_stage.R`.

There is some code that is shared across multiple figures or tables in this repository. For example, `utilities_outcome_experimental_` has helper functions used in several plots. To capture this dependency, there are entries in the Makefile that look like this:

```
analysis/plot_any_exper.R: analysis/utilities_outcome_experimental_plots.R
 touch analysis/plot_any_exper.R
```

The command `touch` here updates the timestamp of `analysis/plot_any_exper.R` which in turn would cause the recipe for `any_exper.pdf` to be re-run, as `plot_any_exper.R` is a dependency.

3.2. Building the Project

To get the data and build the project, you need to obtain a `.env` file from the author and put it in the `~/minimum_wage/writeup` folder. The `.env` file contains: 1) a private URL to the zipped and encrypted data, and 2) a key to decrypt the data. The `.env` file will look like this:

```
GPG_PASSPHRASE='<password>'
project_name="minimum_wage"
DROPBOX_URL="<url of dropbox link hosting the data>"(base)
```

These values are used in a bash script that fetches the data, at fetch_data.sh in the main directory.

There are three options to get the data and build the project: 1) local machine approach, 2) Docker approach, and 3) Replit approach.

3.2.1. Non-Docker, Local Approach

First, assuming the `.env` file is in your `Downloads` folder, run the folloiwng in your command line:

```
git clone git@github.com:johnjosephhorton/minimum_wage.git
cd minimum_wage
cp ~/Downloads/.env writeup
```

Equivalently, you can download the project and manually put the `.env` file in the `writeup` folder.

On a Linux machine, the system_setup.sh file in the main directory will install the right dependencies. In the command line, type:

```
sudo ./system_setup.sh
```

Then, running the following will build the project:

```
cd writeup
make minimum_wage.pdf
```

On a non-Linux machine, ensure the R packages listed in section 4.1.2 are installed on your machine. Then, run the following to build everything:

```
git clone git@github.com:johnjosephhorton/minimum_wage.git
cd minimum_wage/writeup
make minimum_wage.pdf
```

3.2.2. Docker Approach

As in the Non-Docker approach, clone (or download) the repository and ensure a copy of the `.env` file is present in the `writeup` folder. To clone the repository and copy a version of the `.env` file to the `writeup` folder you can type the following in the command line:

```
git clone git@github.com:johnjosephhorton/minimum_wage.git
cd minimum_wage/writeup
cp ~/Downloads/.env .
```

From here, running this will build the entire project:

```
make docker
```

After the project is built, a copy of the final pdf will appear inside the `writeup` folder on your local machine.

3.2.3. Replit Approach

I have created a public replit "repl" here: https://replit.com/@johnhorton/minimumwage. You can **fork** this repository, then add the `.env` file to the `writeup` folder then just push the big green 'Run' button. The dependencies are specified in the replit.nix file available in the main directory. It will generate the PDF and store it in `writeup`.

4. Code, Datasets, and Outputs

This section describes software requirements, datasets used, and outputs of the analysis.

4.1 Program and Software

The entire project has been implemented in R (version 4.4.1). The `analysis` folder contains 23 R scripts that utilize 10 datasets to generate all outputs reported in the paper.

4.1.1. R Scripts

Table below provides a list of all 23 R scripts located in the `analysis` folder.

R Script	R Script
avg_wages_by_cat.R	first_stage.R
jjh_misc.R	parameters.R
parameters_country_selection.R	parameters_effects.R
plot_any_exper.R	plot_application_event_study.R
plot_composition.R	plot_did_all_outcomes.R
plot_event_study_hired_admin.R	plot_event_study_hourly_rate_hired.R
plot_feedback.R	plot_fill_and_hours.R
plot_follow_on_openings.R	plot_hours_zero.R
plot_organic_applications.R	quantile_hours_worked.R
randomization_check.R	realized_wage_distro.R
settings.R	table_any_prior.R
utilities_outcome_experimental_plots.R	

4.1.2. R Packages

The complete list of R packages used in the scripts above, along with the versions I last used to run the code is given in the table below.

R Package (version)	R Package (version)
cowplot (1.1.3)	lmtest (0.9-40)
data.table (1.16.2)	lubridate (1.9.4)
directlabels (2024.1.21)	magrittr (2.0.3)
dplyr (1.1.4)	plyr (1.8.9)
dotenv (1.0.3)	purrr (1.0.2)
ggplot2 (3.5.1)	quantreg (6.00)
ggrepel (0.9.6)	sandwich (3.1-1)
gridExtra (2.3)	scales (1.3.0)
gt (0.11.1)	stargazer (5.1)
lfe (3.1.1)	tidyr (1.3.1)

4.2. Datasets

The analysis uses 10 different datasets, all of which are automatically stored in the `data` folder after initiating the data download procedure (more on this later). Table below gives a list of all datasets along with a brief description of each. The `codebooks` folder contains 10 Markdown files that document the contents of each dataset.

Dataset	Description
df_mw_first.csv	Main experimental outcomes at job post level (first observation)
df_mw_all.csv	Main experimental outcomes at job post level (all observations)
df_mw_admin.csv	Main experimental outcomes at job post level for admin data
df_mw_lpw.csv	Main experimental outcomes at job post level for low wage positions
df_exp_results.csv	Aggregated experimental results and summary statistics
hires_country_composition.csv	Geographic composition of hires
event_study_windows.csv	Time windows and intervals used in the event study analysis
did_panel.csv	Data used for the DiD analysis
event_study_hired.csv	Detailed hiring outcome data for event study analyses
event_study_windows_hr_v_fp.csv	Composition of fixed price and hourly jobs over time

4.3. Outputs

4.3.1. Figures

Table below provides a detailed mapping between each figure used in the text of the paper, which R script generates it, and its dataset dependencies. (Note that the location in paper entries correspond to the published version of the manuscript, not the version produced by the replication package.) All figures will be stored in `writeup/plots` after the project is built.

Figure	R Script	Data Dependency	Location in Paper
Figure 1	analysis/realized_wage_distro.R	data/df_mw_first.csv	Page 125
Figure 2	analysis/first_stage.R	data/df_mw_first.csv	Page 126
Figure 3	None (TikZ Diagram)	None	Page 126
Figure 4	analysis/plot_fill_and_hours.R	data/df_mw_all.csv, df_mw_admin.csv, df_mw_lpw.csv	Page 130
Figure 5	analysis/plot_composition.R	data/df_mw_all.csv, df_mw_admin.csv, df_mw_lpw.csv	Page 134
Figure 6	analysis/plot_event_study_hourly_rate_hired.R	data/event_study_hired.csv	Page 138
Figure 7	analysis/plot_did_all_outcomes.R	data/did_panel.csv	Page 140
Figure 8	analysis/plot_application_event_study.R	data/event_study_windows.csv	Page 143
Figure A1	analysis/plot_organic_applications.R	data/df_mw_all.csv, df_mw_admin.csv, df_mw_lpw.csv	Page A2
Figure A2	analysis/plot_follow_on_openings.R	data/df_mw_all.csv, df_mw_admin.csv, df_mw_lpw.csv	Page A3
Figure A3	analysis/avg_wages_by_cat.R	data/df_mw_first.csv	Page A6
Figure B1	analysis/plot_hours_zero.R	data/df_mw_all.csv, df_mw_admin.csv, df_mw_lpw.csv	Page A7
Figure B2	analysis/plot_any_exper.R	data/df_mw_all.csv, df_mw_admin.csv, df_mw_lpw.csv	Page A9
Figure B3	analysis/plot_feedback.R	data/df_mw_all.csv, df_mw_admin.csv, df_mw_lpw.csv	Page A9
Figure B4	analysis/plot_event_study_hired_admin.R	data/event_study_hired.csv	Page A11

4.3.2. Tables

Table below provides a detailed mapping between each table used in the text of the paper, which R script generates it, and its dataset dependencies. (Note that the location in paper entries correspond to the published version of the manuscript, not the version produced by the replication package.) All tables will be stored in `writeup/tables` after the project is built.

Table	R Script	Data Dependency	Location in Paper
Table A1	analysis/randomization_check.R	data/df_mw_first.csv	Page A2
Table B1	analysis/quantile_hours_worked.R	data/event_study_windows_hr_v_fp.csv	Page A8
Table B2	analysis/table_any_prior.R	data/df_mw_first.csv	Page A10

4.3.3. Parameters

Some of the scripts do not generate tables or figures, but instead generate "parameters." These are numbers that are called out in the text of the paper but are generated automatically. For example, after the project is built the `parameters/parameters.tex` file will contain a line `\newcommand{\numTotal}{159,656}` that is related to the total allocation to the experiment. If we look in `parameters.R` script we see the corresponding code that creates this value.

```
addParam("\\numTotal", formatC(dim(df.mw.first)[1], big.mark = ","))
```

The table below provides the mapping between each parameter file and the R script that generates it. All parameter files are saved in `writeup/parameters` after the corresponding R scripts is run.

Parameter File	R Script
parameters/parameters.tex	analysis/parameters.R
parameters/effects_parameters.tex	analysis/parameters_effects.R
parameters/parameters_fill_and_hours.tex	analysis/plot_fill_and_hours.R
parameters/did_parameters.tex	analysis/plot_did_all_outcomes.R
parameters/parameters_composition.tex	analysis/plot_composition.R
parameters/params_country_selection.tex	analysis/parameters_country_selection.R

4.4. License for Code

The code is licensed under an MIT license.

5. Data Availability and Provenance

5.1. Provenance

This data comes from a large online labor market (Anonymous Online Platform, 2023) that conducted the experiment described in the paper. The data was pulled from the company's database using SQL. The data used for this analysis is proprietary and confidential, but may likely be obtained with Data Use Agreements with the data provider. The data provider itself chooses to remain anonymous. As such, researchers interested in access to the data may contact John Horton at `jjhorton@mit.edu`. The author will assist with any reasonable replication attempts for two years following publication.

5.2. Statement about Rights

- ✓ I certify that the author(s) of the manuscript have legitimate access to and permission to use the data used in this manuscript.
- ❍ I certify that the author(s) of the manuscript have documented permission to redistribute/publish the data contained within this replication package. Appropriate permissions are documented in the LICENSE.txt file in the main directory.

5.3. Summary of Availability

- ❍ All data **are** publicly available.
- ❍ Some data **cannot be made** publicly available.
- ✓ **No data can be made** publicly available.

6. Computational Requirements

The computational requirements for reproducing this paper are minimal. The project can be run on a modern laptop in about 20 minutes, using only open-source softwares.

The code was last run on 2025-03-30 15:16:41. The machine details when I last ran it are:
System info:

- OS: "Ubuntu 20.04.6 LTS"
- Processor: 11th Gen Intel(R) Core(TM) i7-11850H @ 2.50GHz, 16 cores
- Memory available: 31GB memory

The Linux dependencies are mostly either for LaTeX or dependencies for various R packages.

6.1. Runtime

Approximate time needed to reproduce the analyses on a standard 2023 desktop machine is about 20 minutes:

- ❍ <10 minutes

✓ 10–60 minutes

❍ 1–2 hours

❍ 2–8 hours

❍ 8–24 hours

❍ 1–3 days

❍ 3–14 days

❍ >14 days

❍ Not feasible to run on a desktop machine, as described below.

7. References

- Anonymous Online Platform. 2023. "Production Database on Contractors," Unpublished Data, Accessed December 9, 2023.

- Horton, John J. 2025a. "Price Floors and Employer Preferences: Evidence from a Minimum Wage Experiment." American Economic Review 115, No. 1: 117–146. https://doi.org/10.1257/aer.20170637.

- Horton, John J. 2025b. "minimum_wage (Version 1.0)." GitHub Repository. Accessed April 9, 2025. https://github.com/johnjosephhorton/minimum_wage/releases/tag/v1.0.

- Horton, John J. 2025c. "Code For: Price Floors and Employer Preferences: Evidence from a Minimum Wage Experiment." American Economic Association, Inter-university Consortium for Political and Social Research. http://doi.org/10.3886/E208551V1.