Difference-in-Differences Analysis: Minimum Wage Effect

Replication of Card & Krueger (1994) Fast-Food Employment Study

Hursh Gupta Setu Jalandar

2025-04-07

Table of contents

1 Introduction

We will perform a Difference-in-Differences (DiD) analysis to estimate the impact of a minimum wage increase on employment in the fast-food industry.

NOTE: The analysis replicates the classic study by Card and Krueger (1994), which utilized data from fast-food restaurants in New Jersey (treatment group) and Pennsylvania (control group) before and after New Jersey raised its minimum wage.

References:

- Card, David, and Alan B. Krueger. 1994. "Minimum Wages and Employment: A Case Study of the Fast-Food Industry in New Jersey and Pennsylvania." *American Economic Review* 84 (4): 772–93.

- Dataset: http://davidcard.berkeley.edu/data_sets/njmin.zip

2 Setup

First, we load the necessary R packages using `pacman` for package management. We also define a helper function for creating nicely formatted tables and set a default theme for our plots.

2.1 Packages

```
library(pacman)
pacman::p_load(
  tidyverse,  # For data manipulation (dplyr, tidyr) and plotting (ggplot2)
  ggplot2,    # For plotting
  lfe,        # For efficient fixed effects
  skimr,      # For better summary statistics
  stargazer,
  knitr,       # For knitting the notebook and kable tables
)

# Define user-specific path - **ADJUST THIS PATH**
path = "D:/analysis/Econometrics/project/card_kruger/"

# Helper function for nice tables
nice_table <- function(x, ...) {
  knitr::kable(x, digits = 3, ...)
}

# Set default theme for ggplot
theme_set(theme_minimal())

# Optional: uncomment if running in standard R GUI on Windows
# windows()
```

2.2 Retrieving Raw Data

First we will use this function to download the data (credits: aaronmams.github.io)

```r
tempfile_path <- tempfile()
download.file(
  "http://davidcard.berkeley.edu/data_sets/njmin.zip",
  destfile = tempfile_path)
tempdir_path <- tempdir()
unzip(tempfile_path, exdir = tempdir_path)
codebook <- read_lines(file = paste0(tempdir_path, "/codebook"))

variable_names <- codebook %>%
  `[`(8:59) %>%
  `[`(-c(5, 6, 13, 14, 32, 33)) %>%
  str_sub(1, 13) %>%
  str_squish() %>%
  str_to_lower()

dataset <- read_table2(paste0(tempdir_path, "/public.dat"),
                       col_names = FALSE)

dataset <- dataset %>%
  select(-X47) %>%
  `colnames<-`(., variable_names) %>%
  mutate_all(as.numeric) %>%
  mutate(sheet = as.character(sheet)) %>%
  mutate(
    state = ifelse(state == 1, "nj", "pa")
  )


write.csv(dataset,file="data/fast-food-data.csv")
```

Then we load this csv file data into our project in order to perform analysis.

```r
# Load the dataset using read.csv
# Ensure the file exists at the specified 'path'
raw_dat <-  read.csv(file.path(path,'data/fast-food-data.csv'))
```

Let's examine the structure of the raw dataset to understand its variables and format. The original data is often in a "wide" format, with separate columns for measurements taken before and after the policy change.

```r
# Display the structure of the raw data
glimpse(raw_dat)
```

```
Rows: 410
Columns: 47
$ X        <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18~
$ sheet    <int> 46, 49, 506, 56, 61, 62, 445, 451, 455, 458, 462, 468, 469, 4~
$ chain    <int> 1, 2, 2, 4, 4, 4, 1, 1, 2, 2, 3, 1, 1, 1, 1, 2, 2, 3, 3, 3, 3~
$ co_owned <int> 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1~
$ state    <chr> "pa", "pa", "pa", "pa", "pa", "pa", "pa", "pa", "pa", "pa", "~
$ southj   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
$ centralj <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
$ northj   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
$ pa1      <int> 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
$ pa2      <int> 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
$ shore    <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
$ ncalls   <int> 0, 0, 0, 0, 0, 2, 0, 0, 0, 2, 0, 0, 0, 0, 2, 2, 0, 0, 1, 2, 0~
$ empft    <dbl> 30.0, 6.5, 3.0, 20.0, 6.0, 0.0, 50.0, 10.0, 2.0, 2.0, 2.5, 40~
$ emppt    <dbl> 15.0, 6.5, 7.0, 20.0, 26.0, 31.0, 35.0, 17.0, 8.0, 10.0, 20.0~
$ nmgrs    <dbl> 3, 4, 2, 4, 5, 5, 3, 5, 5, 2, 3, 3, 5, 3, 3, 3, 1, 2, 3, 2, 4~
$ wage_st  <dbl> NA, NA, NA, 5.00, 5.50, 5.00, 5.00, 5.00, 5.25, 5.00, 5.00, 5~
$ inctime  <dbl> 19, 26, 13, 26, 52, 26, 26, 52, 13, 19, 13, 13, 39, NA, 26, 2~
$ firstinc <dbl> NA, NA, 0.37, 0.10, 0.15, 0.07, 0.10, 0.25, 0.25, 0.15, 0.37,~
$ bonus    <int> 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0~
$ pctaff   <dbl> NA, NA, 30, 0, 0, 45, 0, 0, 0, 0, 5, 0, 80, 0, 0, 0, 0, 0, 0,~
$ meals    <int> 2, 2, 2, 2, 3, 2, 2, 2, 1, 1, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2~
$ open     <dbl> 6.5, 10.0, 11.0, 10.0, 10.0, 10.0, 6.0, 0.0, 11.0, 11.0, 9.0,~
$ hrsopen  <dbl> 16.5, 13.0, 10.0, 12.0, 12.0, 12.0, 18.0, 24.0, 10.0, 10.0, 1~
$ psoda    <dbl> 1.03, 1.01, 0.95, 0.87, 0.87, 0.87, 1.04, 1.05, 0.73, 0.94, 1~
$ pfry     <dbl> 1.03, 0.90, 0.74, 0.82, 0.77, 0.77, 0.88, 0.84, 0.73, 0.73, 1~
$ pentree  <dbl> 0.52, 2.35, 2.33, 1.79, 1.65, 0.95, 0.94, 0.96, 2.32, 2.32, 1~
$ nregs    <int> 3, 4, 3, 2, 2, 2, 3, 6, 2, 4, 4, 4, 3, 3, 3, 5, 4, 6, 5, 4, 4~
$ nregs11  <int> 3, 3, 3, 2, 2, 2, 3, 4, 2, 4, 4, 3, 2, 3, 1, 4, 3, 3, 5, 4, 4~
$ type2    <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1~
$ status2  <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
$ date2    <int> 111792, 111292, 111292, 111492, 111492, 111492, 110792, 11179~
$ ncalls2  <int> 1, NA, NA, NA, NA, NA, NA, 2, NA, 1, 1, NA, 1, 2, 1, 2, NA, N~
$ empft2   <dbl> 3.5, 0.0, 3.0, 0.0, 28.0, NA, 15.0, 26.0, 3.0, 2.0, 1.0, 9.0,~
$ emppt2   <dbl> 35, 15, 7, 36, 3, NA, 18, 9, 12, 9, 25, 32, 39, 10, 20, 4, 13~
$ nmgrs2   <dbl> 3, 4, 4, 2, 6, NA, 5, 6, 2, 2, 4, 4, 4, 3, 3, 3, 3, 3, 3, 3, ~
$ wage_st2 <dbl> 4.30, 4.45, 5.00, 5.25, 4.75, NA, 4.75, 5.00, 5.00, 5.00, 4.7~
$ inctime2 <int> 26, 13, 19, 26, 13, 26, 26, 26, 13, 13, 13, 26, 41, 13, NA, 2~
```

```
$ firstin2 <dbl> 0.08, 0.05, 0.25, 0.15, 0.15, NA, 0.15, 0.20, 0.25, 0.25, 0.2~
$ special2 <int> 1, 0, NA, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, ~
$ meals2   <int> 2, 2, 1, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2~
$ open2r   <dbl> 6.5, 10.0, 11.0, 10.0, 10.0, 10.0, 6.0, 0.0, 11.0, 11.0, 9.0,~
$ hrsopen2 <dbl> 16.5, 13.0, 11.0, 12.0, 12.0, 12.0, 18.0, 24.0, 11.0, 10.5, 1~
$ psoda2   <dbl> 1.03, 1.01, 0.95, 0.92, 1.01, NA, 1.04, 1.11, 0.94, 0.90, 1.0~
$ pfry2    <dbl> NA, 0.89, 0.74, 0.79, 0.84, 0.84, 0.86, 0.84, 0.84, 0.73, 1.0~
$ pentree2 <dbl> 0.94, 2.35, 2.33, 0.87, 0.95, 1.79, 0.94, 0.94, 2.32, 2.32, 0~
$ nregs2   <int> 4, 4, 4, 2, 2, 3, 3, 6, 4, 4, 6, 4, 3, 3, 3, 3, 4, 6, 5, 6, 5~
$ nregs112 <int> 4, 4, 3, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 2, 2, 3, 3, 1, 3, 3, 3~
```

3 Data Cleaning and Reshaping

The raw data needs to be reshaped from a wide format to a long format, which is more suitable for panel data analysis like DiD. We create a `post` variable (0 for observations before the wage increase, 1 for observations after). We also calculate Full-Time Equivalent (FTE) employment.

3.1 Reshape to Long Format

We separate the data from the two waves (before and after) and stack them, creating the `post` indicator.

```r
# Select pre-policy variables, rename ID,
# mark as pre-period (post=0)
df_pre <- raw_dat %>%
  select(
    store_id = X, # Assuming 'X' is the store identifier column
    chain, state, co_owned, starts_with("empft"), starts_with("emppt"),
    starts_with("nmgrs"), starts_with("wage_st")
    # Add any other time-varying controls if needed
    ) %>%
  # Remove any accidentally included post-policy vars
  select(-ends_with("2")) %>%
  mutate(post = 0)

# Select pre-policy variables, rename ID and
# post-policy  variables, mark as post-period (post=1)
df_post <- raw_dat %>%
  # Select ID, time-invariant vars, and wave 2 vars
```

```r
  select(
    store_id = X,
    chain, state, co_owned, ends_with("2")
  ) %>%
  # Rename wave 2 variables by removing the '2' suffix
  rename_with(~ gsub("2$", "", .x), ends_with("2")) %>%
  mutate(post = 1)

# Combine pre and post dataframes
long_df <- bind_rows(df_pre, df_post)
```

3.2 Final Cleaning and Variable Creation

We convert variables to appropriate types (numeric, factor), calculate FTE employment (`emp_total`), create the `treatment` dummy variable (1 for NJ, 0 for PA), and select the final set of columns for analysis.

```r
# Final data transformations
df <- long_df %>%
  mutate(
    # Ensure key variables are numeric
    empft = as.numeric(empft),
    emppt = as.numeric(emppt),
    wage_st = as.numeric(wage_st),
    nmgrs = as.numeric(nmgrs),
    # Calculate Total Full Time Equivalent employment (FTE)
    emp_total = empft + (0.5 * emppt),
    # Create treatment dummy: 1 if NJ (treatment), 0 if PA (control)
    treatment = ifelse(state == "nj", 1, 0),
    # Convert character variables to factors
    chain = factor(chain),
    state = factor(state)
  ) %>%
  # Select final columns
  select(
    store_id, treatment, post, state, chain, co_owned,
    empft, emppt, emp_total, wage_st, nmgrs
  ) %>%
  # Arrange data for clarity (optional)
  arrange(store_id, post)
```

4 Explore Cleaned Data

Now, let's look at the structure and summary statistics of the final, cleaned dataset (`df`) that we will use for the analysis.

4.1 Structure of Final Data

```
# Display structure of the cleaned, long-format dataframe
print("Structure of final long dataset")
```

```
[1] "Structure of final long dataset"
```

```
glimpse(df)
```

```
Rows: 820
Columns: 11
$ store_id  <int> 1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7, 8, 8, 9, 9, 10, 10~
$ treatment <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
$ post      <dbl> 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, ~
$ state     <fct> pa, pa, pa, pa, pa, pa, pa, pa, pa, pa, pa, pa, pa, pa, pa, ~
$ chain     <fct> 1, 1, 2, 2, 2, 2, 4, 4, 4, 4, 4, 4, 1, 1, 1, 1, 2, 2, 2, 2, ~
$ co_owned  <int> 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, ~
$ empft     <dbl> 30.0, 3.5, 6.5, 0.0, 3.0, 3.0, 20.0, 0.0, 6.0, 28.0, 0.0, NA~
$ emppt     <dbl> 15.0, 35.0, 6.5, 15.0, 7.0, 7.0, 20.0, 36.0, 26.0, 3.0, 31.0~
$ emp_total <dbl> 37.50, 21.00, 9.75, 7.50, 6.50, 6.50, 30.00, 18.00, 19.00, 2~
$ wage_st   <dbl> NA, 4.30, NA, 4.45, NA, 5.00, 5.00, 5.25, 5.50, 4.75, 5.00, ~
$ nmgrs     <dbl> 3, 3, 4, 4, 2, 4, 4, 2, 5, 6, 5, NA, 3, 5, 5, 6, 5, 2, 2, 2,~
```

4.2 Summary Statistics

We use `skimr` to get a detailed summary of the numeric variables:

```
[1] "Summary of Final Long Dataset (Numeric Variables)"
```

variable	missing	mean	sd	p0	p25	p50	p75	p100	hist
treatment	0	0.807	0.395	0.00	1.0	1.0	1.00	1.00	

variable	missing	mean	sd	p0	p25	p50	p75	p100	hist
post	0	0.500	0.500	0.00	0.0	0.5	1.00	1.00	
co_owned	0	0.344	0.475	0.00	0.0	0.0	1.00	1.00	
empft	18	8.239	8.299	0.00	2.0	6.0	12.00	60.00	
emppt	14	18.755	10.387	0.00	11.0	17.0	25.00	60.00	
emp_total	19	17.595	9.023	0.00	11.5	16.5	22.00	80.00	
wage_st	41	4.806	0.358	4.25	4.5	5.0	5.05	6.25	
nmgrs	12	3.452	1.081	0.00	3.0	3.0	4.00	10.00	

5 Compare Means Before and After Policy Change (EDA)

A key part of DiD is observing the trends in the outcome variable (FTE employment) for both the treatment (NJ) and control (PA) groups before and after the policy change. We calculate the average employment and visualize it.

5.1 Calculate Average Employment

```
# Calculate average FTE employment by state and time period
avg_emp_summary <- df %>%
  group_by(state, post) %>%
  summarise(avg_emp = mean(emp_total, na.rm = TRUE), .groups = 'drop')
```

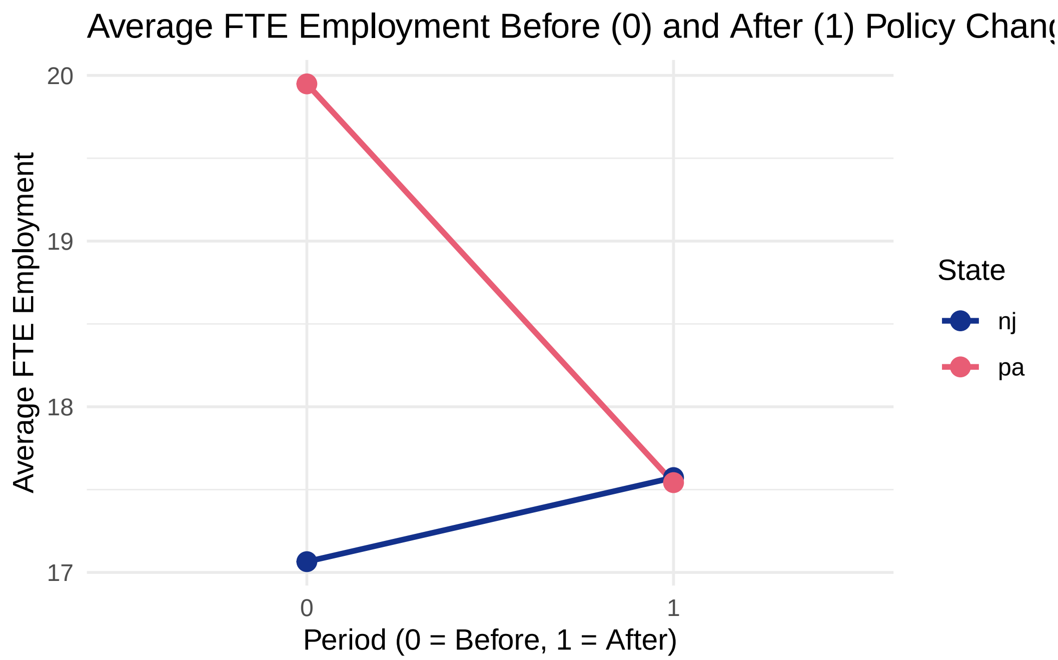
5.2 Plot Average Employment Trends

This plot helps visualize the "difference-in-differences". We compare the change in employment in NJ (blue line) to the change in PA (red line).

```
# Define plot elements
plot_title <- "Average FTE Employment Before (0) and After (1) Policy Change"
emp_plot <- ggplot(
    avg_emp_summary, aes(x = factor(post),
    y = avg_emp, color = state, group = state)
  ) +
  geom_line(linewidth = 1) + # Connect points with lines
  geom_point(size = 3) +     # Show points for means
  labs(
    title = plot_title,
```

```
    x = "Period (0 = Before, 1 = After)",
    y = "Average FTE Employment",
    color = "State"
  ) +
  scale_color_manual(values =
  c("nj" = "#13318C", "pa" = "#E85D75"))

# Display the plot
print(emp_plot)
```



6 Framing the DiD Models

We now estimate the DiD model using regression. The core idea is to model the outcome variable (`emp_total`) as a function of the treatment status (`treatment`), the time period (`post`), and their interaction (`treatment * post`). The coefficient on the interaction term is the DiD estimate of the policy effect.

We estimate several specifications:

1. **Basic DiD:** Only includes treatment, post, and interaction terms.
2. **DiD + Covariates:** Adds control variables (chain, ownership, starting wage, managers).

3. **DiD + Covariates + Fixed Effects (FE):** Adds store-level fixed effects to control for time-invariant unobserved differences between stores.

We use the efficient `feols` function from the `fixest` package.

6.1 Model 1: Basic DiD

```
# Basic DiD model specification
m1 <- lm(emp_total ~ treatment * post, data = df)
```

6.2 Model 2: DiD with Covariates

```
# DiD model including covariates
m2 <- lm(emp_total ~ treatment * post +
    chain + co_owned + wage_st + nmgrs, data = df)
```

6.3 Model 3: DiD with Store Fixed Effects & Covariates

We might want to use the store_id as a fixed effect because there can be a scenario in which say there are two stores, A (from New Jersey) and B(from Pennsylvania). Store A can inherently have **high** employment. Store B can inherently have **low** employment.

```
# DiD model with covariates and store fixed effects (| store_id)
m3 <- lfe::felm(emp_total ~ treatment*post + # DiD
        chain + co_owned + wage_st + nmgrs # Covariates
      | store_id, #Control for
  data = df)
```

```
Warning in chol.default(mat, pivot = TRUE, tol = tol): the matrix is either
rank-deficient or not positive definite
```

7 Conclusion

7.1 Model Results

We present the results from the four models side-by-side using the `modelsummary` package for a clear comparison.

% Table created by stargazer v.5.2.3 by Marek Hlavac, Social Policy Institute. E-mail: marek.hlavac at gmail.com % Date and time: Mon, Apr 07, 2025 - 4:24:15 PM

7.2 Interpretation Notes

The key coefficient of interest in the table above is **`Treatment x Post (DiD Effect)`**. This coefficient estimates the average causal effect of the minimum wage increase on FTE employment in New Jersey *relative to* the change observed in Pennsylvania over the same period.

- A **positive coefficient** suggests employment increased more (or decreased less) in NJ relative to PA after the policy change.
- A **negative coefficient** suggests employment decreased more (or increased less) in NJ relative to PA.

Model 3 controls for all time-invariant unobserved differences between stores (e.g., location, baseline management quality).

We observe that the did coefficient's positive across the three models, hinting that employment actually increased more in NJ relative to PA after the policy change. But is this inference statistically reliable?

Statistical significance suggests the likelihood that the observed did effect is truly different from zero, rather than due to random chance.

> The results in the table are consistent with Card and Krueger's original findings that the minimum wage increase in New Jersey did *not* lead to a significant decrease in fast-food employment compared to Pennsylvania.

Table 2: DiD Estimates of Minimum Wage Effect on FTE Employment

	Dependent variable:		
	emp_total		
	OLS		*felm*
	(1)	(2)	(3)
treatment	-2.884^{**}	-1.772^{*}	
	(1.135)	(1.016)	
post	-2.407^{*}	-1.938	-2.012^{*}
	(1.446)	(1.298)	(1.050)
chain2		-8.801^{***}	
		(0.821)	
chain3		-0.737	
		(0.815)	
chain4		-0.947	
		(0.898)	
co_owned		-1.298^{*}	
		(0.675)	
wage_st		2.601^{**}	2.238^{*}
		(1.049)	(1.309)
nmgrs		1.776^{***}	0.552
		(0.285)	(0.393)
treatment:post	2.914^{*}	1.331	1.188
	(1.611)	(1.527)	(1.337)
Constant	19.949^{***}	3.500	
	(1.019)	(4.998)	
Observations	801	760	760
R^2	0.008	0.248	0.793
Adjusted R^2	0.004	0.239	0.547
Residual Std. Error	9.003 (df = 797)	7.804 (df = 750)	6.017 (df = 347)
F Statistic	2.155^{*} (df = 3; 797)	27.452^{***} (df = 9; 750)	

Note: *p<0.1; **p<0.05; ***p<0.01