Assignment 3

Solutions

Mark Deming

3/02/2025

```
knitr::opts_chunk$set(echo = TRUE, warning = FALSE, message = FALSE)

# Load packages
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(readr)
library(here)
```

```
## here() starts at /Users/mdeming/Library/CloudStorage/Box-Box/r_stuff/card-krueger_1994_replication
```

```
library(modelsummary)
```

```
## `modelsummary` 2.0.0 now uses `tinytable` as its default table-drawing
##   backend. Learn more at: https://vincentarelbundock.github.io/tinytable/
##
## Revert to `kableExtra` for one session:
##
##   options(modelsummary_factory_default = 'kableExtra')
##   options(modelsummary_factory_latex = 'kableExtra')
##   options(modelsummary_factory_html = 'kableExtra')
##
## Silence this message forever:
##
##   config_modelsummary(startup_message = FALSE)
```

```r
library(kableExtra)
```

```
##
## Attaching package: 'kableExtra'
##
## The following object is masked from 'package:dplyr':
##
##     group_rows
```

```r
# Read in data
ck <- read_csv(here("data","original","card-krueger_1994.csv"))
```

```
## Rows: 410 Columns: 46
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (46): sheet, chain, co_owned, state, southj, centralj, northj, pa1, pa2,...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Overview

In this assignment, you will replicate portions of Card & Krueger's analysis of minimum wages and employment (1994). The assignment culminates in a complete replication of Table 4, in which the authors run a series of regressions using a differences-in-differences design (DiD).

Getting started

1. Download the Card & Krueger (1994) data from the course Github site. You should also download the corresponding description of variables.

2. Start by loading the following packages in the `setup` chunk at the top of this RMD:

- tidyverse
- readr
- here
- modelsummary

3. Read in the Card & Krueger (1994) data. The data are in .csv format. Take some time to explore the data:

- structure (dimensions, cross-sectional vs. panel, etc.)
- variables (names, types, etc.)

Cleaning & Transforming

4. The authors create a new variable, full-time equivalent employment (`fte`). They describe the variable on page 775. Follow the authors' definition to create 2 new variables: `fte1` and `fte2`. The variables should measure full-time employment for waves 1 and 2, respectively.

```
ck <- ck %>%
  mutate(fte1 = empft + nmgrs + emppt*0.5,
         fte2 = empft2 + nmgrs2 + emppt2*0.5)
```

5. The values of the variable `chain` denotes different restaurant chains. Use this variable to create 4 new dummy variables, one for each restaurant chain: e.g., `bk` = 1 if a restaurant is a Burger King and 0 otherwise.

```
ck <- ck %>%
  mutate(bk = case_when(chain == 1 ~ 1, chain != 1 ~ 0),
         kfc = case_when(chain == 2 ~ 1, chain != 2 ~ 0),
         roys = case_when(chain == 3 ~ 1, chain != 3 ~ 0),
         wendys = case_when(chain == 4 ~ 1, chain != 4 ~ 0))
```

Table 2

6. Let's ensure that the data match the authors'. To do so, calculate and print the proportions shown in 1a-e, 2a, and 3a of Table 2 on page 776. It is *not* necessary to create a nice table.

```
# Section 1: Chains
(section1 <- ck %>%
  select(state, bk, kfc, roys, wendys, co_owned) %>%
  group_by(state) %>%
  summarize_all(mean) %>%

  # The code above is sufficient for the assignment.
  # The code below refines the dataframe. It is
  # included for reference.

  pivot_longer(cols = -state,
               names_to = "chain",
               values_to = "value") %>%
  pivot_wider(id_cols = "chain",
              names_from = "state",
              values_from = "value"))
```

```
## # A tibble: 5 x 3
##   chain      `0`   `1`
##   <chr>    <dbl> <dbl>
## 1 bk       0.443 0.411
## 2 kfc      0.152 0.205
## 3 roys     0.215 0.248
## 4 wendys   0.190 0.136
## 5 co_owned 0.354 0.341
```

```
# Sections 2 and 3: FTE
(section2 <- ck %>%
  select(state, fte1, fte2) %>%
  group_by(state) %>%
  summarize_all(mean, na.rm = TRUE) %>%
```

```r
  # The code above is sufficient for the assignment.
  # The code below refines the dataframe. It is
  # included for reference.

  pivot_longer(cols = -state,
               names_to = "variable",
               values_to = "value") %>%
  pivot_wider(id_cols = "variable",
              names_from = "state",
              values_from = "value"))
```

```
## # A tibble: 2 x 3
##   variable   `0`   `1`
##   <chr>     <dbl> <dbl>
## 1 fte1       23.3  20.4
## 2 fte2       21.2  21.0
```

Figure 1

7. Use `ggplot` to replicate Figure 1, panel 1 (February 1992). **Hint:** Before creating the plot, you will need to bin `wage_st` into discrete categories. The relevant function is `cut()`.

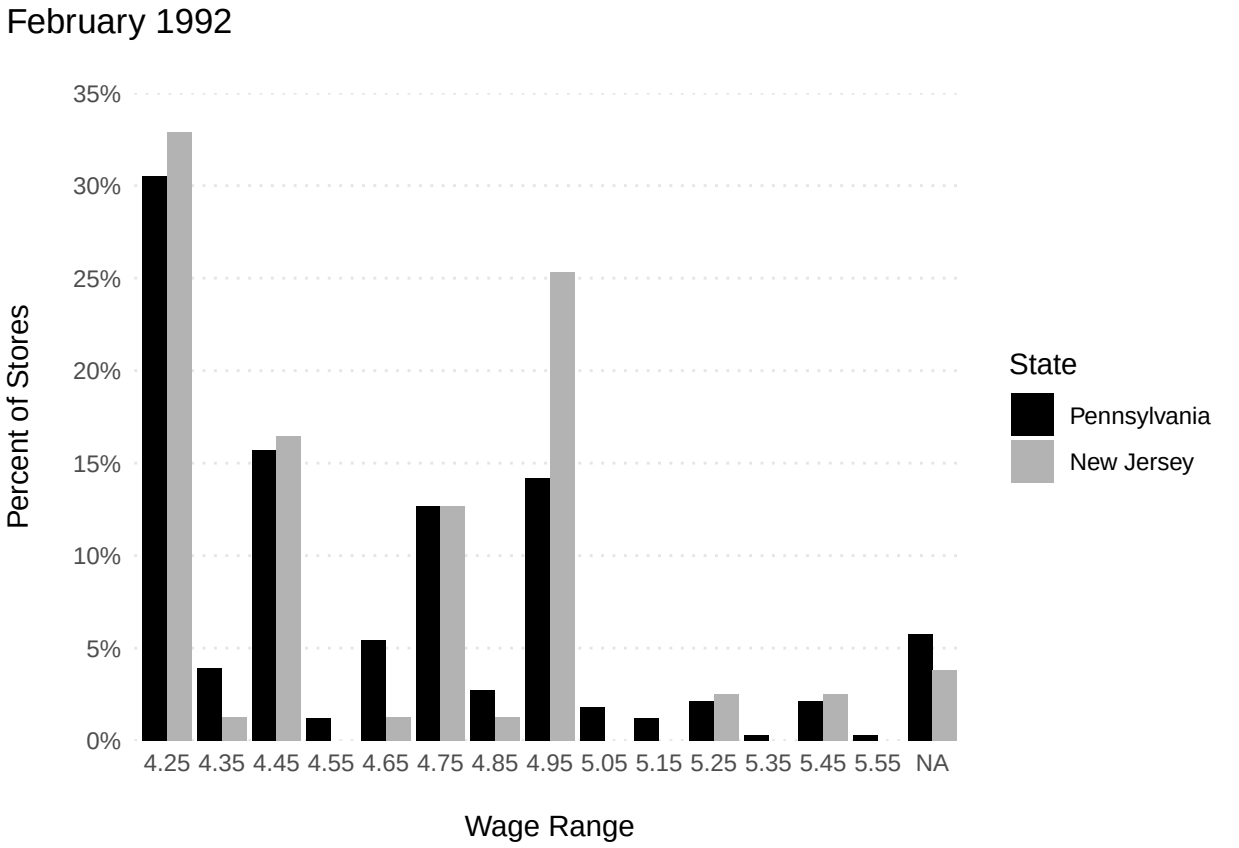
```r
ck <- ck %>%
  mutate(cuts = cut(wage_st, breaks = seq(4.20, 5.6, by = .10),
                             labels = seq(4.25, 5.6, by = .10)))
```

```r
# Basic Feb. 1992 plot
p1 <- ggplot(ck,
             aes(x = cuts,                                    # Plot cuts, not wage
                 group = fct_rev(factor(state)),              # PA should be first
                 fill = fct_rev(factor(state)))) +            # PA should be first
  geom_bar(aes(y = after_stat(prop)),                         # Proportions along y-axis
           position = position_dodge(preserve = "single"))    # Maintain uniform bar width when count =
```

```r
# This code refines the basic plot above. It is not necessary to
# go this far with your own code. I provide the code for reference.

p1 + scale_fill_manual(values = c("black","gray70"),              # Fill color
                       labels = c("Pennsylvania","New Jersey")) + # Legend labels
  scale_y_continuous(limits = c(0, .35),                          # Y-axis limits
                     breaks = seq(0, .35, .05),                   # Y-axis breaks
                     labels = scales::percent,                    # Display Y-axis labels as percent
                     expand = c(0, 0)) +                          # Remove white space
  labs(title = "February 1992\n",                                 # Plot title
       x = "\nWage Range",                                        # X-axis title
       y = "Percent of Stores\n",                                 # Y-axis title
       fill = "State") +                                          # Legend title
  theme_minimal() +                                               # Add "minimal" theme
  theme(panel.grid.major.x = element_blank(),                     # Remove x-axis major grid lines
        panel.grid.major.y = element_line(linetype = 3,           # Adjust y-axis major grid lines
                                          color = "gray90"),
```

```
        panel.grid.minor = element_blank(),                    # Remove all minor grid lines
        plot.title.position = "plot")                          # Push plot title all the way to left
```



8. Use `ggplot` to replicate Figure 1, panel 2 (November 1992). As above, you will need to bin `wage_st2` into discrete categories.

```
ck <- ck %>%
  mutate(cuts2 = cut(wage_st2, breaks = seq(4.20, 5.6, by = .10),
                               labels = seq(4.25, 5.6, by = .10)))
```

```
# Basic Nov. 1992 plot
p2 <- ggplot(ck,
             aes(x = cuts2,
                 group = fct_rev(factor(state)),
                 fill = fct_rev(factor(state)))) +
  geom_bar(aes(y = after_stat(prop)),
           position = position_dodge(preserve = "single"))
```

```
# This code refines the basic plot above. It is not necessary to
# go this far with your own code. I provide the code for reference.

p2 + scale_fill_manual(values = c("black","gray70"),               # Fill color
                       labels = c("Pennsylvania","New Jersey")) + # Legend labels
  scale_y_continuous(limits = c(0, .95),                           # Y-axis limits
                     breaks = seq(0, .95, .05),                    # Y-axis breaks
```

```
                    labels = scales::percent,                    # Display Y-axis labels as percent
                    expand = c(0, 0)) +                          # Remove white space
  labs(title = "November 1992\n",                                # Plot title
       x = "\nWage Range",                                       # X-axis title
       y = "Percent of Stores\n",                                # Y-axis title
       fill = "State") +                                         # Legend title
  theme_minimal() +                                              # Add "minimal" theme
  theme(panel.grid.major.x = element_blank(),                    # Remove x-axis major grid lines
        panel.grid.major.y = element_line(linetype = 3,          # Adjust y-axis major grid lines
                                          color = "gray90"),
        panel.grid.minor = element_blank(),                      # Remove all minor grid lines
        plot.title.position = "plot")                            # Push plot title all the way to left
```

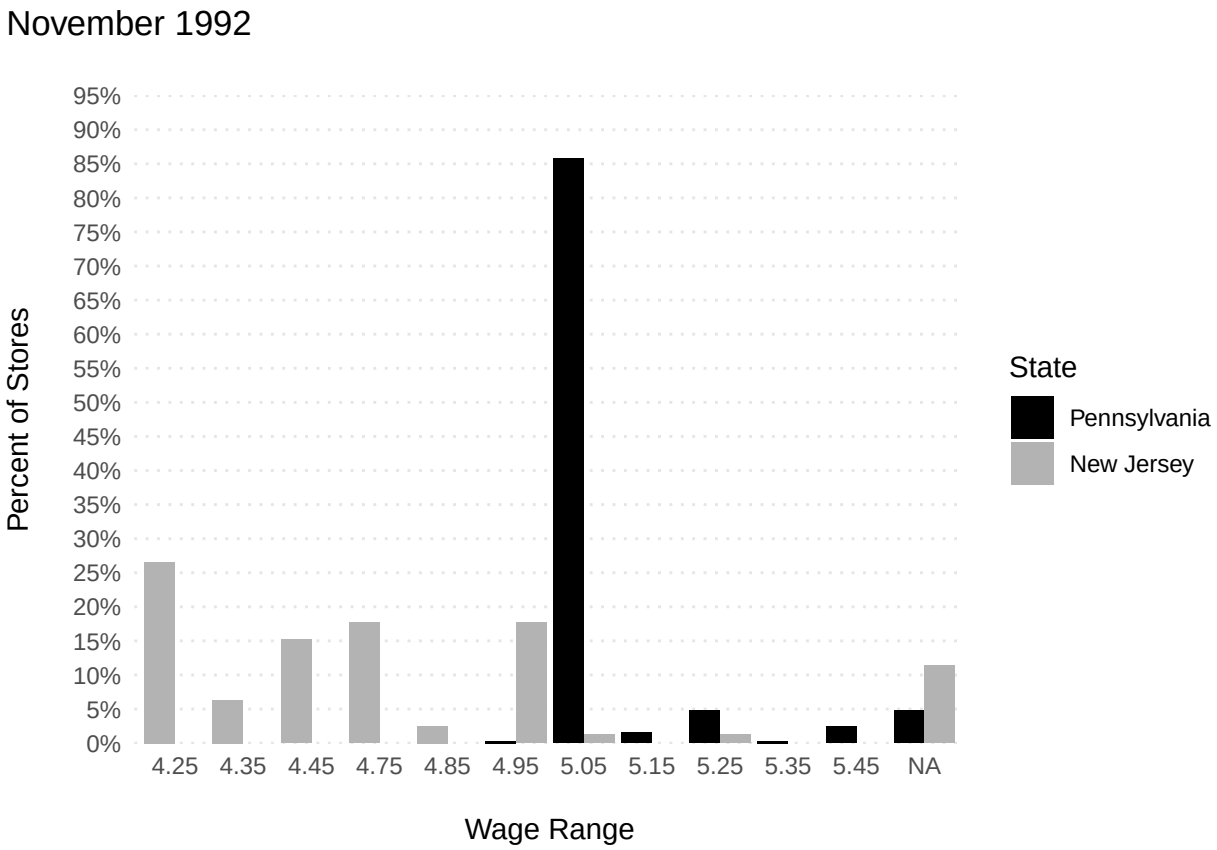


Table 4

9. You will now replicate the authors' DiD design in Table 4. To get started, create a new variable named `gap`. Use the variable definition given by the authors on page 779. **Hint:** You will need to use 2 logical conditions inside `ifelse()`.

```
# Create new "gap" variable
ck <- ck %>%
  mutate(gap=ifelse(state ==1 & wage_st <= 5.05,((5.05 - wage_st) / wage_st), 0))
```

10. Filter the dataset to include only (1) rows with complete data for `fte`, `fte2`, `wage_st`, and `wage_st2` OR (2) rows for restaurants that closed in wave 2.

```
# Select complete observations
ck <- ck %>%
  filter(complete.cases(fte1, fte2)) %>%
  filter(complete.cases(wage_st, wage_st2) | status2 == 3)
```

11. Replicate Table 4, models I through V (p. 780). Specifically, write 5 regression models. Compare the model output against Table 4. It should be the same for models I-IV and very similar for Model V.

```
# Model 1
mod1 <- lm(fte2 - fte1 ~ state, data = ck)

# Model 2
mod2 <- lm(fte2 - fte1 ~ state + I(bk) + I(kfc) + I(roys) + I(wendys) + I(co_owned), data = ck)

# Model 3
mod3 <- lm(fte2 - fte1 ~ gap, data = ck)

# Model 4
mod4 <- lm(fte2 - fte1 ~ gap + I(bk) + I(kfc) + I(roys) + I(wendys) + I(co_owned), data = ck)

# Model 5
mod5 <- lm(fte2 - fte1 ~ gap + I(bk) + I(kfc) + I(roys) + I(wendys) + I(co_owned) +
                         I(centralj) + I(southj) + I(pa1) + I(pa2), data = ck)
```

12. Use `modelsummary()` to print a nice regression table of models I-V above.

```
models <- list(mod1, mod2, mod3, mod4, mod5)

rows <- tibble::tribble(~IV, ~i, ~ii, ~iii, ~iv, ~v,
                        "Controls for chain and ownership", "no", "yes", "no", "yes", "yes",
                        "Controls for region", "no", "no", "no", "no", "yes")
attr(rows, "position") <- c(5, 6)

(table4 <- modelsummary(models,
                        stars = T,
                        title = "Reduced-form models for change in employment",
                        coef_map = c("state" = "New Jersey dummy",
                                     "gap" = "Initial wage gap"),
                        gof_omit = "DF|Deviance|R2|AIC|BIC|Log.Lik|F",
                        add_rows = rows,
                        ouput = "markdown"))
```

13. From a technical perspective, DiD is simple. Why is it a credible design despite its simplicity? Write your answer in a few sentences below.

Table 1: Reduced-form models for change in employment

	(1)	(2)	(3)	(4)	(5)
New Jersey dummy	2.326+	2.304+			
	(1.192)	(1.196)			
Initial wage gap			15.653*	14.916*	11.979
			(6.080)	(6.205)	(7.419)
Controls for chain and ownership	no	yes	no	yes	yes
Controls for region	no	no	no	no	yes
Num.Obs.	357	357	357	357	357
RMSE	8.77	8.71	8.73	8.69	8.63

+ p <0.1, * p <0.05, ** p <0.01, *** p <0.001