Replication of Minimum Wage and Employment by Card and Krueger

Contents

Preparing your workfile

R is a powerful software for statistical analysis. It is open source and hence FREE software. It is constantly developped and functionality is being improved. The "price" we pay for this is that we have to do a little more set-up work to make it work.

In particular we need packages which are provided for free by researchers/programmers that provide useful functionality.

But we add libraries which enhance its capabilities.

```r
library(tidyverse)     # for almost all data handling tasks
library(readxl)        # to import Excel data
library(ggplot2)       # to produce nice graphiscs
library(stargazer)     # to produce nice results tables
```

Introduction

Here we will replicate a seminal piece of work by David Card and Alan B. Krueger, Minimum Wage and Employment: A Case Study of the Fast-Food Industry in New Jersey and Pennsylvania, AER, 1994.

The data and the code in STATA are available from the data page of Angrist and Pischke's Mostly Harmless Econometrics.

Statistical concepts used

The following statistical concepts are used in this project

- Summary statistics (means)

- Difference-in-Difference (Diff-in-Diff, DiD)

R programming concepts used

The following R programming concepts were used in this document. The links lead to pages that provide help with these issues.

- use of packages/libraries: ECLR
- importing of data: csv.read, read_excel
- Summarising data: descriptive statistics
- Plotting data using ggplot: Cookbook for R, free 1st chapter of datacamp interactive tutorial
- running regressions: ECLR
- using the stargazer package for summary statistics and regresison output
- using the tidverse to produce subsets and groupings of data and summarise these
- hypothesis tests on sample means, using the t.test function

Importing Data

The data here are stored as an xlsx format in `public.xlsx`. In order to import these we require the `readxl` package which we loaded earlier.

```
# make sure you use the correct path
CKdata<- read_xlsx("C:/Rcode/RforQM/Data_Introduction/CK_public.xlsx")
str(CKdata)  # prints some basic info on variables
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    410 obs. of  46 variables:
##  $ SHEET   : num  46 49 506 56 61 62 445 451 455 458 ...
##  $ CHAIN   : num  1 2 2 4 4 4 1 1 2 2 ...
##  $ CO_OWNED: num  0 0 1 1 1 1 0 0 1 1 ...
##  $ STATE   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ SOUTHJ  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ CENTRALJ: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ NORTHJ  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ PA1     : num  1 1 1 1 1 1 0 0 0 1 ...
##  $ PA2     : num  0 0 0 0 0 0 1 1 1 0 ...
##  $ SHORE   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ NCALLS  : num  0 0 0 0 0 2 0 0 0 2 ...
##  $ EMPFT   : chr  "30" "6.5" "3" "20" ...
##  $ EMPPT   : chr  "15" "6.5" "7" "20" ...
##  $ NMGRS   : chr  "3" "4" "2" "4" ...
##  $ WAGE_ST : chr  "." "." "." "5" ...
##  $ INCTIME : chr  "19" "26" "13" "26" ...
##  $ FIRSTINC: chr  "." "." "0.37" "0.1" ...
##  $ BONUS   : num  1 0 0 1 1 0 0 0 0 0 ...
##  $ PCTAFF  : chr  "." "." "30" "0" ...
##  $ MEALS   : num  2 2 2 2 3 2 2 2 1 1 ...
##  $ OPEN    : num  6.5 10 11 10 10 10 6 0 11 11 ...
##  $ HRSOPEN : num  16.5 13 10 12 12 12 18 24 10 10 ...
##  $ PSODA   : chr  "1.03" "1.01" "0.95" "0.87" ...
##  $ PFRY    : chr  "1.03" "0.9" "0.74" "0.82" ...
##  $ PENTREE : chr  "0.52" "2.35" "2.33" "1.79" ...
##  $ NREGS   : chr  "3" "4" "3" "2" ...
##  $ NREGS11 : chr  "3" "3" "3" "2" ...
##  $ TYPE2   : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ STATUS2 : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ DATE2   : num  111792 111292 111292 111492 111492 ...
##  $ NCALLS2 : chr  "1" "." "." "." ...
##  $ EMPFT2  : chr  "3.5" "0" "3" "0" ...
##  $ EMPPT2  : chr  "35" "15" "7" "36" ...
##  $ NMGRS2  : chr  "3" "4" "4" "2" ...
##  $ WAGE_ST2: chr  "4.3" "4.45" "5" "5.25" ...
##  $ INCTIME2: chr  "26" "13" "19" "26" ...
##  $ FIRSTIN2: chr  "0.08" "0.05" "0.25" "0.15" ...
##  $ SPECIAL2: chr  "1" "0" "." "0" ...
##  $ MEALS2  : chr  "2" "2" "1" "2" ...
##  $ OPEN2R  : chr  "6.5" "10" "11" "10" ...
##  $ HRSOPEN2: chr  "16.5" "13" "11" "12" ...
##  $ PSODA2  : chr  "1.03" "1.01" "0.95" "0.92" ...
##  $ PFRY2   : chr  "." "0.89" "0.74" "0.79" ...
##  $ PENTREE2: chr  "0.94" "2.35" "2.33" "0.87" ...
##  $ NREGS2  : chr  "4" "4" "4" "2" ...
##  $ NREGS112: chr  "4" "4" "3" "2" ...
```

Not all the variable names are intuitive. The codebook contains details regarding the variables. Importantly you can see which data format the variables have. In particular you can see numeric (`num`) and character/text (`chr`) variables. As you can see, even the variables which are labeld as `chr` are actually numbers. The reason for this is that there are some missing observations (`"."`). As we use the `read_xlsx` function to import the data we can specify that missing data are coded with an “.”. By specifying this at the outset we ensure that the data are formatted as numerical data where appropriate.

```r
CKdata<- read_xlsx("C:/Rcode/RforQM/Data_Introduction/CK_public.xlsx",na = ".")
str(CKdata)  # prints some basic info on variables
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    410 obs. of  46 variables:
##  $ SHEET   : num  46 49 506 56 61 62 445 451 455 458 ...
##  $ CHAIN   : num  1 2 2 4 4 4 1 1 2 2 ...
##  $ CO_OWNED: num  0 0 1 1 1 1 0 0 1 1 ...
##  $ STATE   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ SOUTHJ  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ CENTRALJ: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ NORTHJ  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ PA1     : num  1 1 1 1 1 1 0 0 0 1 ...
##  $ PA2     : num  0 0 0 0 0 0 1 1 1 0 ...
##  $ SHORE   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ NCALLS  : num  0 0 0 0 0 2 0 0 0 2 ...
##  $ EMPFT   : num  30 6.5 3 20 6 0 50 10 2 2 ...
##  $ EMPPT   : num  15 6.5 7 20 26 31 35 17 8 10 ...
##  $ NMGRS   : num  3 4 2 4 5 5 3 5 5 2 ...
##  $ WAGE_ST : num  NA NA NA 5 5.5 5 5 5 5.25 5 ...
##  $ INCTIME : num  19 26 13 26 52 26 26 52 13 19 ...
##  $ FIRSTINC: num  NA NA 0.37 0.1 0.15 0.07 0.1 0.25 0.25 0.15 ...
##  $ BONUS   : num  1 0 0 1 1 0 0 0 0 0 ...
##  $ PCTAFF  : num  NA NA 30 0 0 45 0 0 0 0 ...
##  $ MEALS   : num  2 2 2 2 3 2 2 2 1 1 ...
##  $ OPEN    : num  6.5 10 11 10 10 10 6 0 11 11 ...
##  $ HRSOPEN : num  16.5 13 10 12 12 12 18 24 10 10 ...
##  $ PSODA   : num  1.03 1.01 0.95 0.87 0.87 0.87 1.04 1.05 0.73 0.94 ...
##  $ PFRY    : num  1.03 0.9 0.74 0.82 0.77 0.77 0.88 0.84 0.73 0.73 ...
##  $ PENTREE : num  0.52 2.35 2.33 1.79 1.65 0.95 0.94 0.96 2.32 2.32 ...
##  $ NREGS   : num  3 4 3 2 2 2 3 6 2 4 ...
##  $ NREGS11 : num  3 3 3 2 2 2 3 4 2 4 ...
##  $ TYPE2   : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ STATUS2 : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ DATE2   : num  111792 111292 111292 111492 111492 ...
##  $ NCALLS2 : num  1 NA NA NA NA NA NA 2 NA 1 ...
##  $ EMPFT2  : num  3.5 0 3 0 28 NA 15 26 3 2 ...
##  $ EMPPT2  : num  35 15 7 36 3 NA 18 9 12 9 ...
##  $ NMGRS2  : num  3 4 4 2 6 NA 5 6 2 2 ...
##  $ WAGE_ST2: num  4.3 4.45 5 5.25 4.75 NA 4.75 5 5 5 ...
##  $ INCTIME2: num  26 13 19 26 13 26 26 26 13 13 ...
##  $ FIRSTIN2: num  0.08 0.05 0.25 0.15 0.15 NA 0.15 0.2 0.25 0.25 ...
##  $ SPECIAL2: num  1 0 NA 0 0 0 0 0 0 0 ...
##  $ MEALS2  : num  2 2 1 2 2 2 2 2 2 1 ...
##  $ OPEN2R  : num  6.5 10 11 10 10 10 6 0 11 11 ...
##  $ HRSOPEN2: num  16.5 13 11 12 12 12 18 24 11 10.5 ...
##  $ PSODA2  : num  1.03 1.01 0.95 0.92 1.01 NA 1.04 1.11 0.94 0.9 ...
##  $ PFRY2   : num  NA 0.89 0.74 0.79 0.84 0.84 0.86 0.84 0.84 0.73 ...
##  $ PENTREE2: num  0.94 2.35 2.33 0.87 0.95 1.79 0.94 0.94 2.32 2.32 ...
##  $ NREGS2  : num  4 4 4 2 2 3 3 6 4 4 ...
##  $ NREGS112: num  4 4 3 2 2 3 3 3 3 3 ...
```

We can see that now all variables are numeric (`num`) variables which is as we expect.

There are 410 observations, each representing one fast-food restaurant. Each restaurant has some variables which characterise the store and then two sets of variables which are observed before the new minimum wage was introduced to New Jersey (Wave 1: Feb 15 to Mar 14, 1992) and after the policy change (Wave 1: Nov 5 to Dec 31, 1992). Variables which relate to the second wave will have a `2` at the end of the variable name.

The following variables will be important for the analysis here:

- `STATE`, 1 if New Jersey (NJ); 0 if Pennsylvania (Pa)
- `WAGE_ST`, starting wage ($/hr), Wave 1
- `WAGE_ST2`, starting wage ($/hr), after policy, Wave 2
- `STATUS2`, the status of the Wave 2 interview, 0 = refused, 1 = completed, 3, permanently closed, 2, 4 and 5 = temporarily closed
- `CHAIN`, 1 = Burger King; 2 = KFC; 3 = Roy Rogers; 4 = Wendy's
- `EMPFT`, # full-time employees before policy implementation
- `EMPFT2`, # full-time employees after policy implementation
- `EMPPT`, # part-time employees before policy implementation
- `EMPPT2`, # part-time employees after policy implementation
- `NMGRS`, # managers/ass't managers before policy implementation
- `NMGRS2`, # managers/ass't managers before policy implementation

For later it will be convenient to have `STATE` and `CHAIN` variables which aren't numeric, but categorical variables. In R these are called `factor` variables.

```
CKdata$STATEf <- as.factor(CKdata$STATE)  # translates a variable into a factor variable
levels(CKdata$STATEf) <- c("Pennsylvania","New Jersey") # changes the names of the categories

CKdata$CHAINf <- as.factor(CKdata$CHAIN)
levels(CKdata$CHAINf) <- c("Burger King","KFC", "Roy Rogers", "Wendy's")
```

Some Summary Statistics

Let's create some summary statistics, replicating elements of Card and Krueger's Table 1.

```
Tab1 <- CKdata %>% group_by(STATEf) %>%
          summarise(n = n()) %>%
          print()
```

```
## # A tibble: 2 x 2
##   STATEf           n
##   <fct>        <int>
## 1 Pennsylvania    79
## 2 New Jersey     331
```

Let's exlpain what the command does. It uses the powerful syntax introduced by the `tidyverse` (which is why we needed loading this package). This is a super useful technique and you can find a more detailed introduction on ECLR. But let's translate the above command into English to illustrtae what it does.

We create a new object named `Tab1`. To that new object we assign the result of the operation on the right of `<-`. So what happens to the right of `<-`? Start with object `CKdata` (wich is our entire spreadsheet with all observations and all variables) and send it (the piping command `%>%`) to the `group_by` function. That function does what t says, it groups the data but it requires some additional information, i.e. by what variable it should group. We are asking it to group by the `STATEf` variable, i.e. group the data (or split the data) into a subset for each state (here PA and NJ). Then send (`%>%`) the result (the grouped dataset) to the `summarise` function. In here we create a new variable (`n =`) which takes the value `n()`. That last bit makes no obvious sense unless you know (and now you do) that `n()` is the function called `n` which counts the number of observations. So it will count the observations for the the grouped datasets. Then we send (`%>%`) the result to the `print()` function, which eventually produces some output.

Try yourself to change some of that function. For instance replace `STATEf` with `CHAINf` and see what happens. Or, even more interesting, replace `STATEf` with `CHAINf,STATEf` and sit back in awe. How long would it have taken you in Excel to do that?

A frequency table for the `STATUS2` variable we can obtain as follows (note that `STATE = 1` for New Jersey):

```
table(CKdata$STATUS2,CKdata$STATEf)
```

```
##
##     Pennsylvania New Jersey
##   0            0          1
##   1           78        321
##   2            0          2
##   3            1          5
##   4            0          1
##   5            0          1
```

All these data are identical to those in the Card and Krueger paper.

Now we replicate some of the summary statistics in Table 2. First we want proportions of different chain types. At core of this we will first calculate a frequency table again (`table()`) but then we feed the result of this straight into the `prop.table()` function which translates frequencies into proportions. The addition of the `margin = 2` option ensures that proportions are calculated by state (2=columns). Try for yourself what changes if you either set `margin = 1` (1 for rows) or leave this option out.

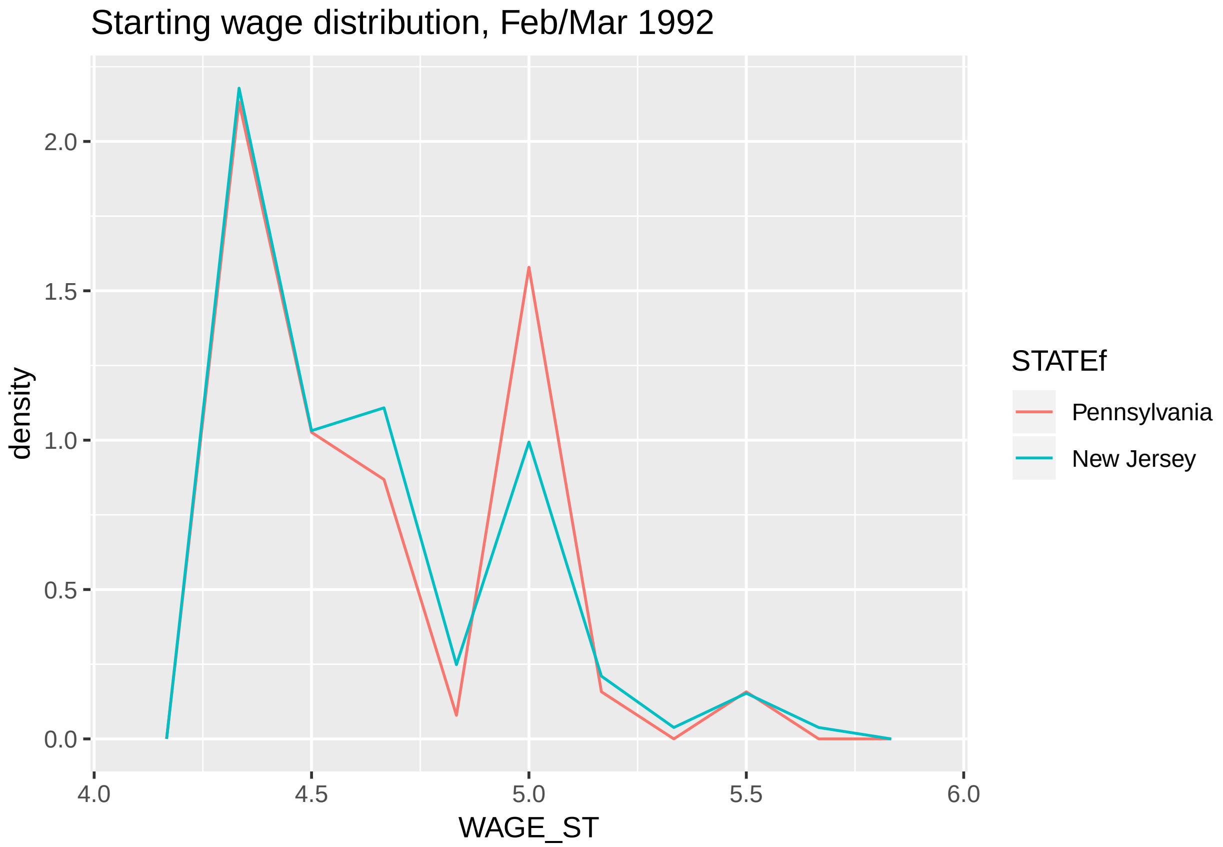
```
prop.table(table(CKdata$CHAINf,CKdata$STATEf,dnn = c("Chain", "State")),margin = 2)
```

```
##               State
## Chain          Pennsylvania New Jersey
##   Burger King     0.4430380  0.4108761
##   KFC              0.1518987  0.2054381
##   Roy Rogers       0.2151899  0.2477341
##   Wendy's          0.1898734  0.1359517
```

Let's also see whether there are other differences between the characteristics. For instance we can look at the distribution of starting wages before the change in minimum wage in New Jersey (`WAGE_ST`).

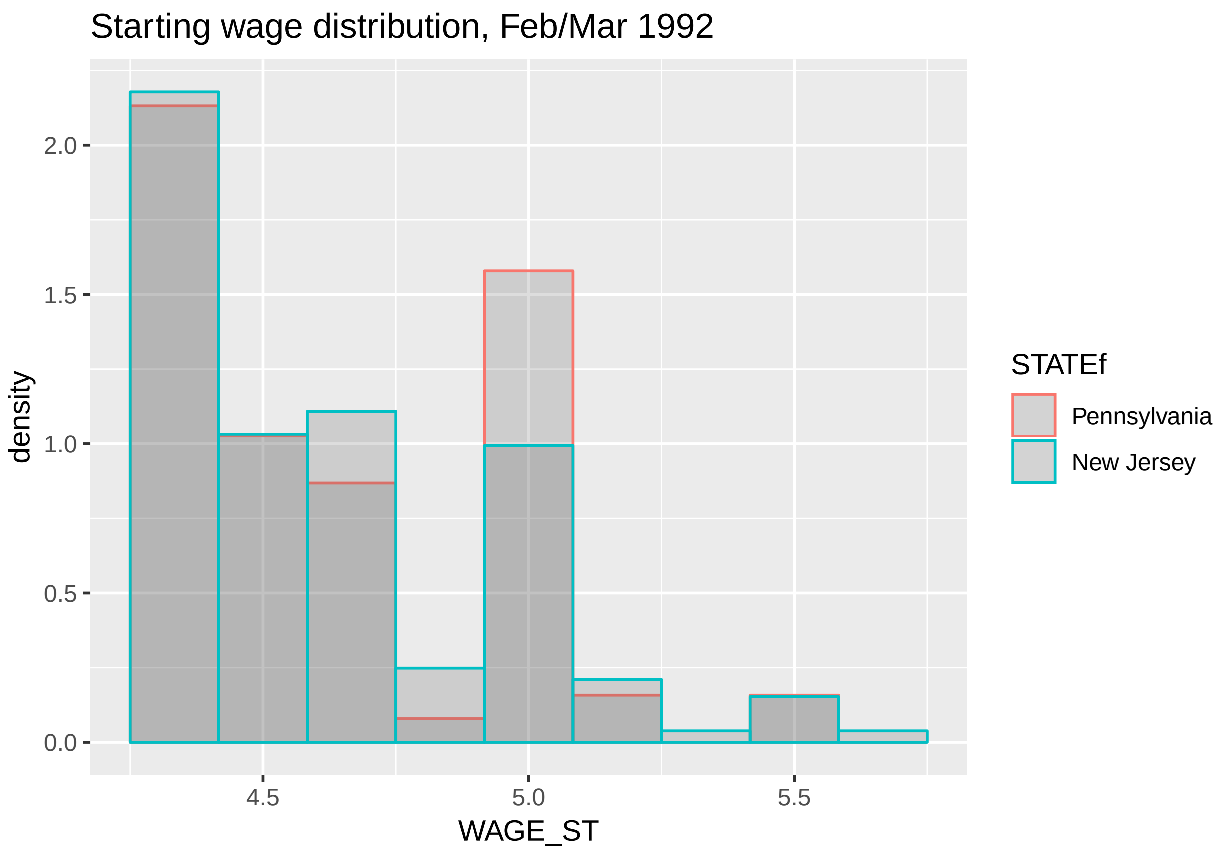
At this stage it is not so important to understand the commands for these plots.

```
ggplot(CKdata, aes(WAGE_ST, stat(density), colour = STATEf)) +
  geom_freqpoly(bins = 10) +
  ggtitle(paste("Starting wage distribution, Feb/Mar 1992"))
```



Or here an alternative visualisation.

```
ggplot(CKdata,aes(WAGE_ST, colour = STATEf), colour = STATEf) +
    geom_histogram(position="identity",
                   aes(y = ..density..),
                   bins = 10,
                   alpha = 0.2) +
    ggtitle(paste("Starting wage distribution, Feb/Mar 1992"))
```



Both plots sow that the starting wage distribution is fairly similar in both states, with peaks at the minimum wage of $4.25 and $5.00.

Policy Evaluation

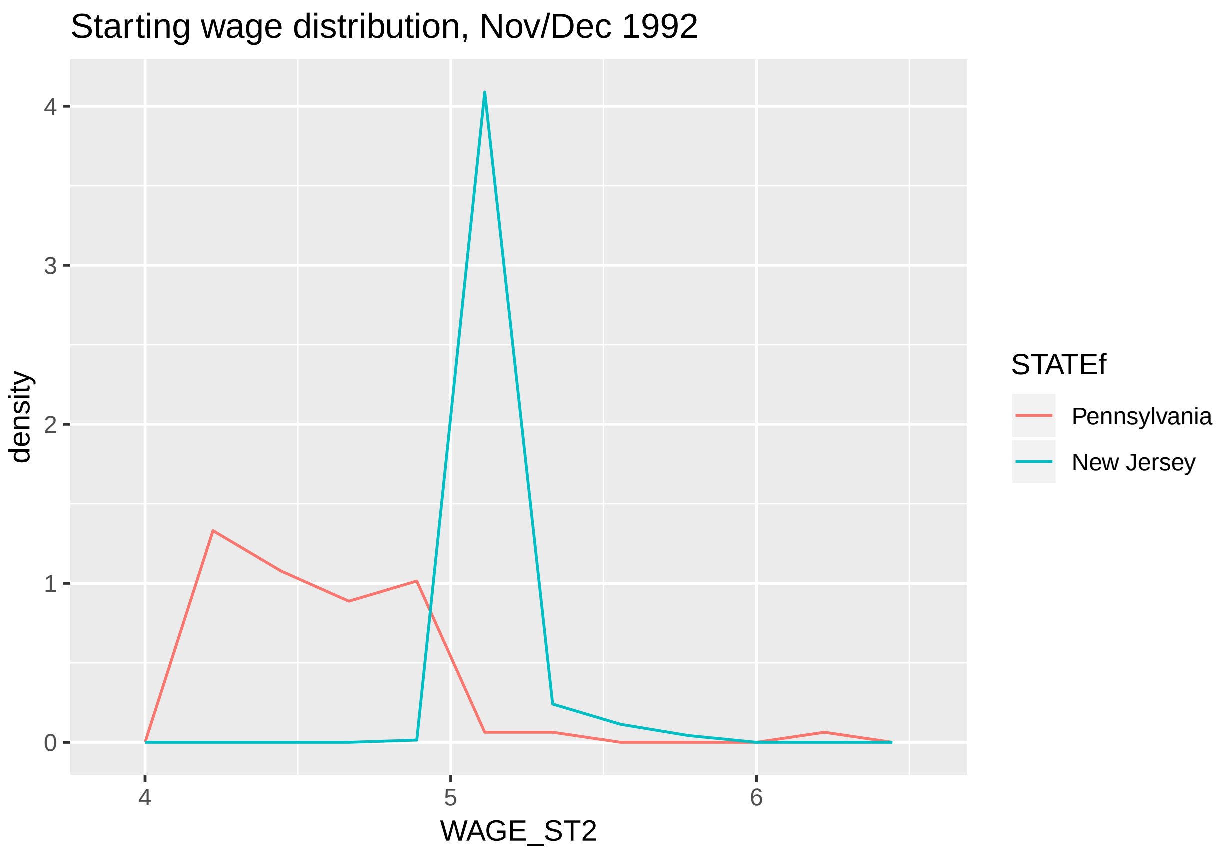
First we can evaluate whether the legislation has been implemented.

```
Tab1 <- CKdata %>% group_by(STATEf) %>%
          summarise(wage_FEB = mean(WAGE_ST,na.rm = TRUE),
                    wage_DEC = mean(WAGE_ST2,na.rm = TRUE)) %>%
          print()
```

```
## # A tibble: 2 x 3
##   STATEf       wage_FEB wage_DEC
##   <fct>           <dbl>    <dbl>
## 1 Pennsylvania     4.63     4.62
## 2 New Jersey       4.61     5.08
```

We can clearly see that the average wage in New Jersey has increased. We could also compare the wage distributions as above.

```
ggplot(CKdata, aes(WAGE_ST2, stat(density), colour = STATEf)) +
  geom_freqpoly(bins = 10) +
  ggtitle(paste("Starting wage distribution, Nov/Dec 1992"))
```



The difference is very obvious.

In order to evaluate whether the increased minimum wage has an impact on employment we want to compare the employment numbers in the two states, before and after the policy implementation.

In the list of variables above you can see that we have before and after policy employee numbers for full-time staff and part-time staff. Card and Krueger calculated a full-time equivalent (FTE) employee number. In order to calculate this they made the assumption that, on average, part-time employees worked 50% of a full-time employee and that manager (`NMGRS` and `NMGRS2`) worked full time.

Hence we will generate two new variables `FTE` and `FTE2`. As almost always the same result can be achieved in different ways in R. So we demonstrate two fifferent ways to create these two variables.

```
CKdata$FTE <- CKdata$EMPFT + CKdata$NMGRS + 0.5*CKdata$EMPPT
CKdata <- CKdata %>%  mutate(FTE2 = EMPFT2 + NMGRS2 + 0.5*EMPPT2)
```

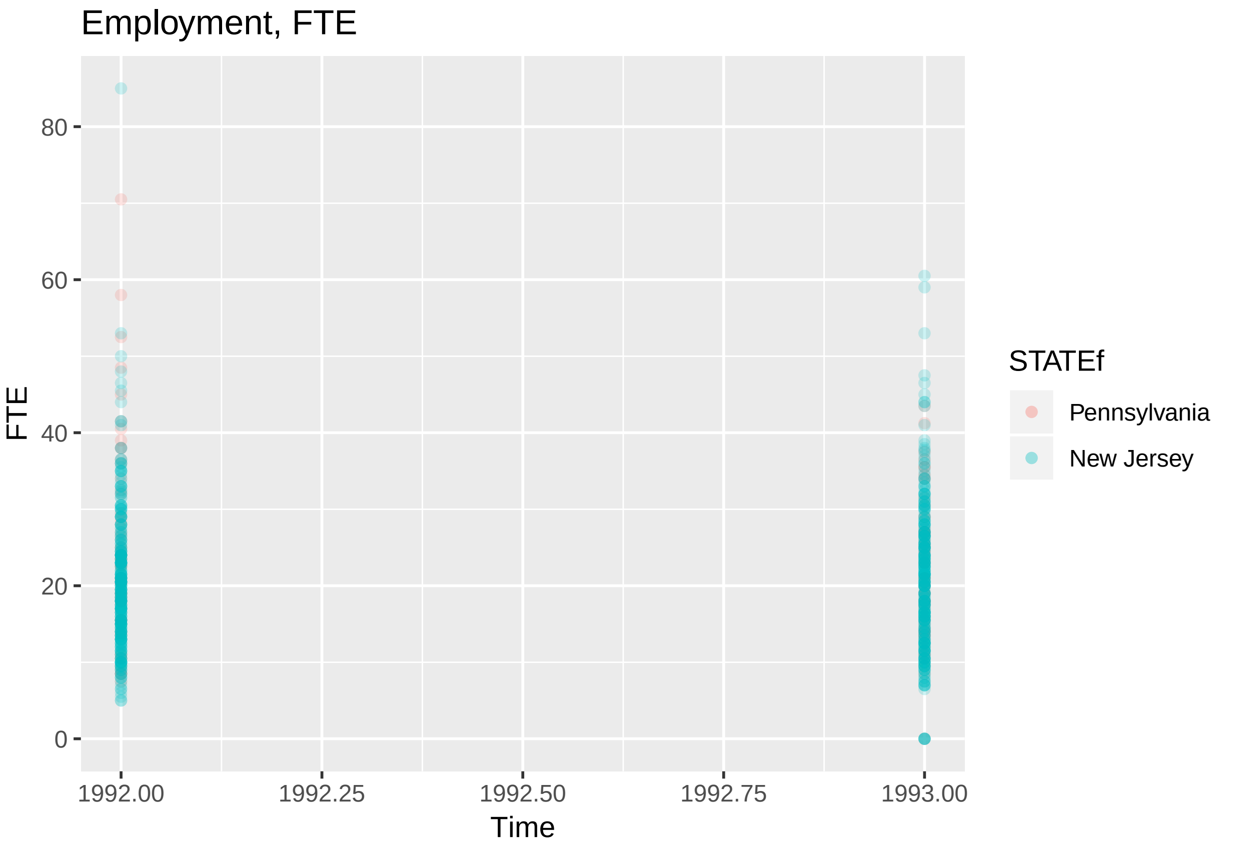
```
TabDiD <- CKdata %>% group_by(STATEf) %>%
          summarise(meanFTE_FEB = mean(FTE,na.rm = TRUE),
                    meanFTE_DEC = mean(FTE2,na.rm = TRUE)) %>%
          print()
```

```
## # A tibble: 2 x 3
##   STATEf       meanFTE_FEB meanFTE_DEC
##   <fct>              <dbl>       <dbl>
## 1 Pennsylvania        23.3        21.2
## 2 New Jersey          20.4        21.0
```

From here you can clearly see that on average stores in New Jersey have increased employment, while the average employment in stores in Pennsylvania has actually decreased. That employment would increase despite the minimum wage increasing was a truly earth-shattering result in 1992.

Let's illustrate the data graphically. We start by plotting all the `FTE` datapoints.

```
ggplot(CKdata, aes(x=1992,y=FTE, colour = STATEf)) +
  geom_point(alpha = 0.2) +
  geom_point(aes(1993,FTE2),alpha = 0.2) +
  labs(x = "Time") +
  ggtitle(paste("Employment, FTE"))
```

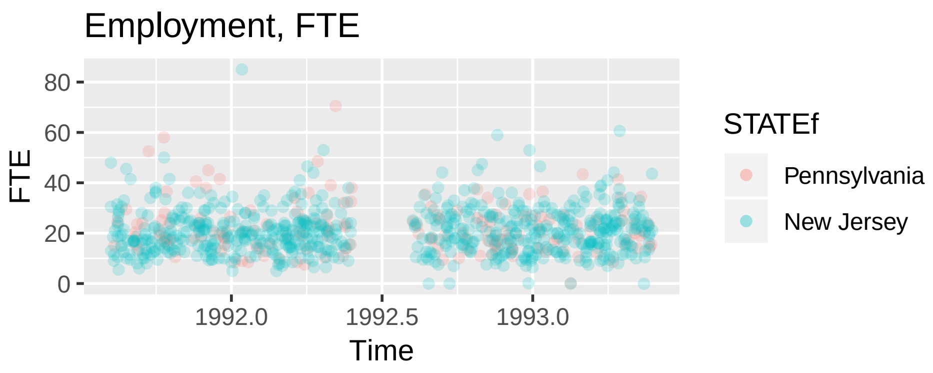


Note a few points about how this graph is being called. `ggplot(CKdata, aes(x=1992,y=FTE, colour = STATEf))` Sets up the graph. The first input (`CKdata`) tells R which data to use. In the `aes()` section (`aes` for aesthetics) we determine which variable should appear on the x-axis (`x=1992`) and which is to go onto the y-axis (`y=FTE`). At this stage we havn't actually produced a graph yet, we just did the ground work. Then we add the graph we want, here a scatterplot, in R `geom_point()`. As you can see, the first line and the second line are linked with a `+`. This has to come at the end of the first line so that R knows that it should expect more information. The next line adds the points for after the increase in the NJ minimum wage (and we assign the time 1993 to these points (Also note that we left out the `x=` and `y=` prefixes. By default R knows that the first input is the x axis variable and the 2nd is the y-axis input). The last line (`ggtitle(paste("Employment, FTE"))`) adds a title.

I havn't mentioned what the `alpha = 0.2` in the `geom_point` commands do. Try and figure this out for yourself either by googling ("R ggplot alpha") or by varying the value of `alpha` in the interval $[0, 1]$ and observe what happens (yes, like in physics experiments … with the added bonus that you cannot break anything!).

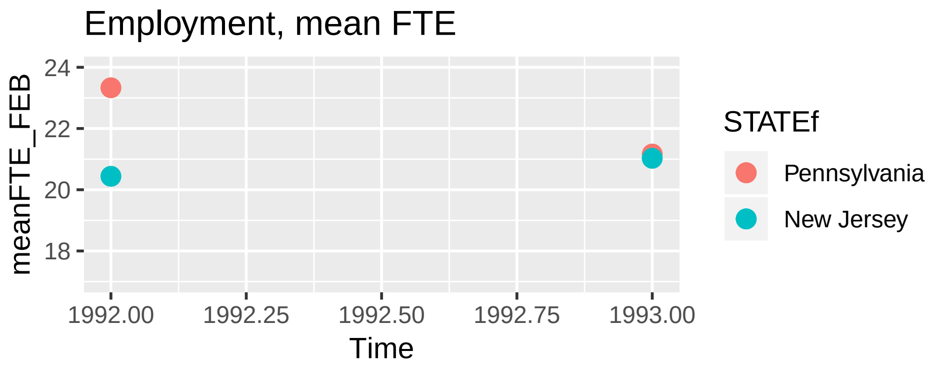
The points are very tight together which make it a little difficult to see how many there really are. Instead of the `geom_point` we use the `geom_jitter` command which jitters the points slightly in the time direction.

```r
ggplot(CKdata, aes(1992,FTE, colour = STATEf)) +
  geom_jitter(alpha = 0.2) +
  geom_jitter(aes(1993,FTE2),alpha = 0.2) +
  labs(x = "Time") +
  ggtitle(paste("Employment, FTE"))
```



When we calculate a DiD estimator we are basically calculating the means of the two staes in the two periods, i.e. four points. We calculated these four values for `TabDiD` and we can plot them. They basically summarise the four clouds of points in the previous picture.

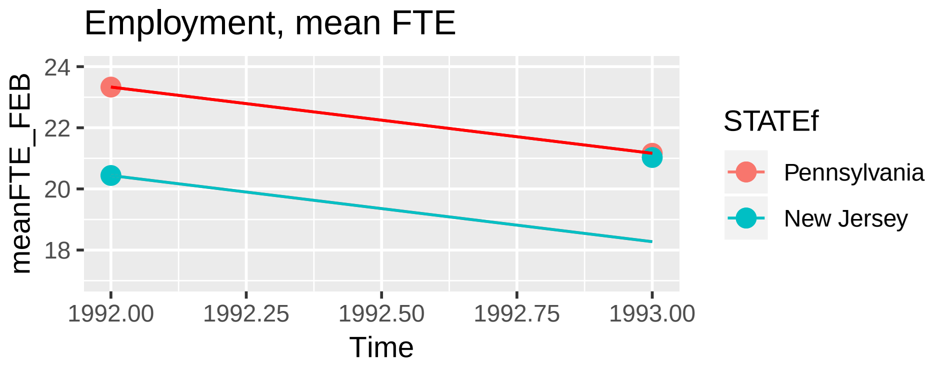
```r
ggplot(TabDiD, aes(1992,meanFTE_FEB, colour = STATEf)) +
  geom_point(size = 3) +
  geom_point(aes(1993,meanFTE_DEC),size=3) +
  ylim(17, 24) +
  labs(x = "Time") +
  ggtitle(paste("Employment, mean FTE"))
```



A DiD analysis now proceeds by assuming that in the treatment state, here New Jersey (NJ), in absence of the policy change the same trend would have prevailed as in the control state, here Pennsylvania (PA). This is a crucial assumption and needs to be made with care.

We will graphically impose the PA trend onto NJ.

```r
ggplot(TabDiD, aes(1992,meanFTE_FEB, colour = STATEf)) +
  geom_point(size = 3) +
  geom_point(aes(1993,meanFTE_DEC),size=3) +
  ylim(17, 24) +
  geom_segment(aes(x = 1992, y = TabDiD[[1,2]],
                   xend = 1993, yend = TabDiD[[1,3]]), color = "red") +
  geom_segment(aes(x = 1992, y = TabDiD[[2,2]],
                    xend = 1993, yend = TabDiD[[2,2]]+(TabDiD[[1,3]]-TabDiD[[1,2]]))) +
  labs(x = "Time") +
  ggtitle(paste("Employment, mean FTE"))
```



The estimated effect of the policy is the distance between the blue dot (the 1993 NJ observation and the end of the blue line). In fact we can use the information in `TabDiD` to obtain the size of the effect.

```
print(TabDiD)
```

```
## # A tibble: 2 x 3
##   STATEf       meanFTE_FEB meanFTE_DEC
##   <fct>              <dbl>       <dbl>
## 1 Pennsylvania        23.3        21.2
## 2 New Jersey          20.4        21.0
```

Numerically the DiD estimator is calculated as follows:

(21 - 20.4) - (21.2 - 23.3) = 2.7

This can be calculated using a regression approach which has the advantage that we can also allow for some additional heterogeneity between observations (here fast food restaurants) and that we can evaluate the statistical significance of the effect size.

But fundamentally, DiD compares just four values, as easy as that.

Further thoughts

These findings clearly contradicted a simplistic analysis of labour markets (assuming they were competitive) which would have suggested that an increased minimum wage should reduce employment. This particular empirical finding was, perhaps not surprisingly, disputed.

- These findings were challenged in Neumark, David, and William Wascher. 2000. “Minimum Wages and Employment: A Case Study of the Fast-Food Industry in New Jersey and Pennsylvania: Comment.” American Economic Review, 90 (5): 1362-1396.
- A reply (by and large defending the original findings) to this criticism was published in the same journal edition by Card, David, and Alan B. Krueger. 2000. “Minimum Wages and Employment: A Case Study of the Fast-Food Industry in New Jersey and Pennsylvania: Reply.” American Economic Review, 90 (5): 1397-1420.
- An overview of the mpirical evidence is provided in this Report by Arindrajit Dube for the UK Government. “Especially for the set of studies that consider broad groups of workers, the overall evidence base suggests an employment impact of close to zero.”