

# Introduction to Handling Data

ECON20222 - Lecture 1

Ralf Becker and Martyn Andrews

# What is this course unit about?

- Help you implement and interpret the main estimation and inference techniques used in Economics
- Focus on:
  - ▶ causal inference
  - ▶ the main pitfalls of time-series analysis

# This Week's Empirical Question



Card, David ; Krueger, Alan B. (1994) Minimum Wages and Employment: A Case Study of the Fast-Food Industry in New Jersey and Pennsylvania, *The American Economic Review*, 84, 772-793.

Do higher minimum wages decrease employment (as predicted by common-sense and a competitive labour market model)?

# The Research Question

“This paper presents new evidence on the effect of minimum wages on establishment-level employment outcomes. We analyze the experiences of 410 fast-food restaurants in New Jersey and Pennsylvania following the increase in New Jersey’s minimum wage from \$ 4.25 to \$ 5.05 per hour. Comparisons of employment, wages, and prices at stores in New Jersey and Pennsylvania before and after the rise offer a simple method for evaluating the effects of the minimum wage.”

Card, David ; Krueger, Alan B. (1994, p.772)

# Why Data Matter

The debate is still alive:

- Overall negative effect on employment, [IZA](#).  
*"Research findings are not unanimous, but especially for the US, evidence suggests that minimum wages reduce the jobs available to low-skill workers."*
- An overview of the empirical 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."*

## At the end of this unit ...

You will be able to:

- Understand and discuss the challenges of making causal inferences
- Perform inference appropriate for the model being estimated
- Interpret empirical results (with due caution!)
- Discuss strengths and weaknesses of particular empirical applications
- Do intermediate data work in R
- Confidently apply regression analysis in R
- Apply more advanced causal inference techniques in R
- Find coding help for any new challenges in R

# What you need to do

To learn in this unit you need to:



coding, cleaning data, struggling,  
self-learning, amazement at what  
you can do

answering real questions, that there  
is not always a clear answer

# Assessment Structure and feedback

- Online test (on the use of R) - 10%
- End-of-Term exam (short answer questions) - 50%
- Group coursework - 40% (see extra info)



# Aim for today

## Statistics/Econometrics

- Summary Statistics
- Difference between population and sample
- Hypothesis testing
- Graphical Data Representations
- Diff-in-Diff Analysis
- Simple regression analysis

## R Coding

- Introduce you to R and RStudio
- How do I learn R
- Import data into R
- Perform some basic data manipulation
- Perform hypothesis tests
- Estimate a regression

# This Week's Plan

- Replicate some of the basic results presented in Card and Krueger (1994)
- Introduce the Difference-in-Difference methodology (Project!!)  
[Sometimes known as “Diff-in-Diff” or DiD.]
- Use this example to
  - ▶ introduce you to R
  - ▶ review some summary statistics
  - ▶ review simple regression and its implementation
  - ▶ introduce some basic visualisations

# Introduce R/R-Studio



- R is a statistical software package, it is open source and free
- a lot of useful functionality is added by independent researchers via packages (also for free)
- RStudio is a user interface which makes working with R easier. You need to install R before you [install RStudio](#).
- [ECLR](#) is a web-resource we have set up to support you in your R work.

# Welcome to RStudio

The screenshot displays the RStudio interface with the following components:

- Source Editor:** Contains R code for data manipulation using `group_by` and `summarize` functions. The code calculates mean and median growth rates for different debt categories.
- Environment Pane:** Shows the global environment with variables like `acadv.sel`, `active_staff`, `admin`, `admin.sel`, `mscdi`, `mscdi.sel`, `phd.sel`, `phdsu`, `staff`, `temp`, `units`, and `values`.
- Console:** Displays the output of the R code, including a table of debt categories and their corresponding mean and median growth rates.
- Terminal:** Shows the output of the `system` function, which runs a command to generate a PDF document.

**Table of Debt Categories and Growth Rates:**

debtcat	n	mean	median
(0,30]	13	4.08921971	4.001282
(30,60]	15	2.86594921	2.889572
(60,90]	14	3.39943999	2.857815
(90,Inf]	7	-0.02421961	1.028900

**Console Output:**

```
$ error : log1 FALSE
$ message: log1 FALSE
$ results: log1 FALSE
$ warning: log1 FALSE

-- Attaching packages ----- tidyverse 1.2.1 --
v ggplot2 2.2.1 v purrr 0.2.4
v tidbldr 1.4.2 v dplyr 0.7.4
v tidy 0.8.0 v stringr 1.3.0
v readr 1.1.1 v forcats 0.3.0

-- Conflicts ----- tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag() masks stats::lag()

Please cite as:
Hlavac, Marek (2018). stargazer: Well-Formatted Regression and Summary Statistics Tables.
R package version 5.2.2. https://CRAN.R-project.org/package=stargazer

[...ordinary text without R code... 100%

"C:/Program Files/RStudio/bin/pandoc/pandoc" +RTS -K512m -RTS Lecture_1_Data_Introduction.utf8.md --to beamer --from markdown+autolink_bare_uris+asciil_identifiers+math_single+backslash --output Lecture_1_Data_Introduction.tex --highlight-style tango --latex-engine pdfLatex --self-contained --include-in-header "\LaTeX_Template.tex"
output file: Lecture_1_Data_Introduction.knit.md

Output created: Lecture_1_Data_Introduction.pdf
warning messages:
1: package 'tidyverse' was built under R version 3.5.1
2: package 'readr' was built under R version 3.5.1
```

## Write Code Files or the Basic Workflow

- keep an original data file (usually ‘.xlsx’ or ‘.csv’) and do not overwrite this file
- any manipulation we make to the data (data cleaning, statistical analysis etc.) is command based and we collect all these commands in a script file. R will then interpret and execute these commands. It is hence like a recipe which you present to a chef. These script files have extension ‘.r’
- you can also learn to write Rmarkdown files (‘.rmd’). They combine code with normal text and output.
- When you write code you should ensure that you add comments to your code. Comments are bit of text which is ignored by R (everything after an ‘#’) but helps you or someone else to decipher what the code does.

By following the above advice you make it easy for yourself and others to replicate your work.

# Prepare your code

We start by uploading the extra packages we need in our code.

The first time you need these packages at a computer you may need to install these. Use the following code to do this

```
install.packages(c("readxl", "tidyverse", "ggplot2", "stargazer"))
```

This only needs to be done once on a particular computer. However, every time you want to use any of these packages in a code you need to make them available to your code (load them):

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

# The data

Then we load the data from excel

```
CKdata<- read_xlsx("CK_public.xlsx",na = ".")
```

na = "." indicates how missing data are coded.

Check some characteristics of the data which are now stored in CKdata:  
Discuss data.frame, number of obs and number of variables, their names and variable types

```
str(CKdata)  # prints some basic info on variables
```

```
## tibble[,46] [410 x 46] (S3: tbl_df/tbl/data.frame)
## $ SHEET      : num [1:410] 46 49 506 56 61 62 445 451 455 458 ...
## $ CHAIN      : num [1:410] 1 2 2 4 4 4 1 1 2 2 ...
## $ CO_OWNED   : num [1:410] 0 0 1 1 1 1 0 0 1 1 ...
## $ STATE      : num [1:410] 0 0 0 0 0 0 0 0 0 0 ...
## $ SOUTHJ     : num [1:410] 0 0 0 0 0 0 0 0 0 0 ...
## $ CENTRALJ   : num [1:410] 0 0 0 0 0 0 0 0 0 0 ...
## $ NORTHJ     : num [1:410] 0 0 0 0 0 0 0 0 0 0 ...
## $ PA1        : num [1:410] 1 1 1 1 1 1 0 0 0 1 ...
## $ PA2        : num [1:410] 0 0 0 0 0 0 1 1 1 0 ...
## $ SHORE      : num [1:410] 0 0 0 0 0 0 0 0 0 0 ...
## $ NCALLS     : num [1:410] 0 0 0 0 0 0 2 0 0 0 2
```

# The data

To see the entire dataset (like in a spreadsheet):

Either click the little spreadsheet symbol next to the data.frame in the Environment tab, or

```
view(CKdata)  # prints some basic info on variables
```



## The data - Unit of observation

A unit of observation is a fast food restaurant.

Say observation 27 in our dataset is a Roy Rogers (`CHAIN = 3`) store in Pennsylvania (`STATE = 0`) with 7 full time employees (`EMPFT`), 19 part-time employees (`EMPPT`) and 4 managers (`NMGRS`) in Feb 1992 and 17.5 in Dec

```
CKdata[27,]
```

```
## # A tibble: 1 x 46
##   SHEET CHAIN CO_OWNED STATE SOUTHJ CENTRALJ NORTHJ   PA1
##   <dbl> <dbl>    <dbl> <dbl>  <dbl>    <dbl> <dbl> <dbl> <
## 1   515     3        1     0      0        0     0     0
## # ... with 35 more variables: EMPFT <dbl>, EMPPT <dbl>, NMGRS <dbl>,
## #   WAGE_ST <dbl>, INCTIME <dbl>, FIRSTINC <dbl>, BONUS <dbl>,
## #   MEALS <dbl>, OPEN <dbl>, HRSOPEN <dbl>, PSODA <dbl>, PF <dbl>,
## #   PENTREE <dbl>, NREGS <dbl>, NREGS11 <dbl>, TYPE2 <dbl>,
## #   DATE2 <dbl>, NCALLS2 <dbl>, EMPFT2 <dbl>, EMPPT2 <dbl>,
## #   WAGE_ST2 <dbl>, INCTIME2 <dbl>, FIRSTIN2 <dbl>, SPECIAL <dbl>
```

## Addressing particular variables

If you want to call/use the entire spreadsheet/data frame/tibble then you call `CKdata`.

But often you want to call one variable only:

- `CKdata$CHAIN`, calls `CHAIN` only
- `CKdata["CHAIN"]`, calls `CHAIN` only
- `CKdata[2]`, calls `CHAIN` only, as it is the 2nd variable

And sometimes you want to call several, but not all, variables:

- `CKdata[c("STATE", "CHAIN")]`

`c("STATE", "CHAIN")` creates a list of names. `c` really represents a function, `c` for concatenation.

**Also note:** `R` is case sensitive, `CHAIN`  $\neq$  `Chain`

# Variable types

These are five basic data types.

- character: `"a", "swc"`
- numeric: `2, 15.5`
- integer: `2L` (the `L` tells `R` to store this as an integer)
- logical: `TRUE, FALSE`
- factor: a set number of categories

It is important that you know and understand differences between data types. Each variable has a particular type and some operations only work for particular datatypes. For instance, we need `num` or `int` for any mathematical operations.

In our `data.frame` we have only `num` variable types.

We will encounter `logical` variables frequently. **they are very powerful**

## factor variables

We store categorical variables as **factor** variables.

Sometimes you need to type convert to **factor** variables.

```
str(CKdata[c("STATE", "CHAIN")]) # prints some basic info on v
```

```
## tibble[,2] [410 x 2] (S3: tbl_df/tbl/data.frame)
```

```
## $ STATE: num [1:410] 0 0 0 0 0 0 0 0 0 0 ...
```

```
## $ CHAIN: num [1:410] 1 2 2 4 4 4 1 1 2 2 ...
```

- STATE, 1 if New Jersey (NJ); 0 if Pennsylvania (Pa)
- CHAIN, 1 = Burger King; 2 = KFC; 3 = Roy Rogers; 4 = Wendy's

## factor variables

```
CKdata$STATEf <- as.factor(CKdata$STATE)
levels(CKdata$STATEf) <- c("Pennsylvania", "New Jersey")

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

- CKdata\$STATE calls variable STATE in dataframe ck\_data
- <- assigns what is on the right as.factor(CKdata\$STATE) to the variable on the left CKdata\$STATEf
- as.factor(CKdata\$STATE) calls a function as.factor and applies it to CKdata\$STATE

```
str(CKdata[c("STATEf", "CHAINf")]) # prints some basic info on variables
```

```
## tibble[,2] [410 x 2] (S3: tbl_df/tbl/data.frame)
## $ STATEf: Factor w/ 2 levels "Pennsylvania",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ CHAINf: Factor w/ 4 levels "Burger King",...: 1 2 2 4 4 4 1 1 2 2 ...
```

## factor variables

`factor` variables are variables with discrete categories. Which ones they are you can find out with the `levels()` function:

```
levels(CKdata$CHAINf)
```

```
## [1] "Burger King" "KFC"          "Roy Rogers"  "Wendy's"
```

## Learn more about your data

Use the `summary` function for some initial summary stats for `num` or `int` variables

- `WAGE_ST`, starting wage (\$/hr), Wave 1, before min wage increase, Feb 1992
- `EMPFT`, # full-time employees before policy implementation

```
summary(CKdata[c("WAGE_ST", "EMPFT")])
```

##	WAGE_ST	EMPFT
##	Min. :4.250	Min. : 0.000
##	1st Qu.:4.250	1st Qu.: 2.000
##	Median :4.500	Median : 6.000
##	Mean :4.616	Mean : 8.203
##	3rd Qu.:4.950	3rd Qu.:12.000
##	Max. :5.750	Max. :60.000
##	NA's :20	NA's :6

# Learn more about your data

How many obs in each state and what chains

```
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
```

```
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
```

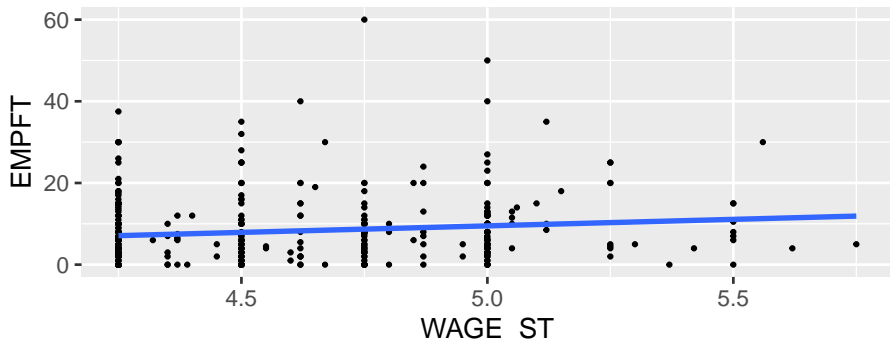
```
## Roy Rogers     0.2151899  0.2477341
```

```
## Wendy's        0.1898734  0.1359517
```



## Scatter plot of the data

```
p1 <- ggplot(CKdata,aes(WAGE_ST,EMPFT)) +  
  geom_point(size=0.5) +      # this produces the scatter plot  
  geom_smooth(method = "lm", se = FALSE) # adds the line  
p1
```



Point out that each dot represents one store data. Point out line of best fit

# Regression Line

The line in the previous plot is the line of best fit coming from a linear regression

$$EMPFT = \alpha + \beta WAGE\_ST + u \text{ (Population Model)}$$

- The population model is defined by unknown parameters  $\alpha$  and  $\beta$  and the unknown error terms  $u$ . We will use sample data to obtain sample estimates of these parameters.
- The error terms  $u$  contain the effects of any omitted variables and reflect that any modelled relationship will only be an approximation. The  $u$  are random variables

$$EMPFT_{it} = \hat{\alpha} + \hat{\beta} WAGE\_ST_{it} + \hat{u}_{it} \text{ (Estimated Sample Model)}$$

Here we have two subscripts as the data have a cross-section (**i**) and a time-series dimension (**t**).

The regression line in the previous figure is represented by

$$\widehat{EMPFT}_{it} = \hat{\alpha} + \hat{\beta} WAGE\_ST_{it} \text{ (Regression Line)}$$

# Simple Regression Model and OLS

Regression analysis is the core technique used in Econometrics. It is based on certain assumptions about the *Population Model* and the error terms  $u$  (more on this in the next few weeks).

How to estimate parameters (get  $\hat{\alpha}$  and  $\hat{\beta}$ ) using the available sample of data? This is typically done by Ordinary Least Squares (OLS).

# Simple Regression Model and OLS

```
mod1 <- lm(EMPFT~WAGE_ST, data= CKdata)
summary(mod1)
```

```
##
## Call:
## lm(formula = EMPFT ~ WAGE_ST, data = CKdata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.091  -5.898  -2.100   3.005  51.304
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -6.468     5.807  -1.114   0.2660
## WAGE_ST         3.193     1.255   2.544   0.0114 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.5 on 383 degrees of freedom
## (25 observations deleted due to missingness)
## Multiple R-squared:  0.01662,    Adjusted R-squared:  0.01405
## F-statistic: 6.472 on 1 and 383 DF,  p-value: 0.01135
```

# OLS - nice output

```
stargazer(mod1,type="text")
```

```
##
## =====
##                               Dependent variable:
##                               -----
##                               EMPFT
## -----
## WAGE_ST                      3.193**
##                               (1.255)
##
## Constant                     -6.468
##                               (5.807)
##
## -----
## Observations                 385
## R2                           0.017
## Adjusted R2                  0.014
## Residual Std. Error         8.500 (df = 383)
## F Statistic                  6.472** (df = 1; 383)
## =====
## Note:                        *p<0.1; **p<0.05; ***p<0.01
```

# OLS - calculation and interpretation

How were  $\hat{\beta}$  and  $\hat{\alpha}$  calculated?

$$\hat{\beta} = \frac{\widehat{Cov}(EMPFT_{it}, WAGE\_ST_{it})}{\widehat{Var}(WAGE\_ST_{it})}$$
$$\hat{\alpha} = \overline{EMPFT}_{it} - \hat{\beta} * \overline{WAGE\_ST}_{it}$$

How to interpret  $\hat{\beta} = 3.193$ ?

An increase of one unit in **WAGE\_ST** (=USD1) is related to an increase in about 3 full time employees (**EMPFT**).

Have we established that higher wages **cause** higher employment?

NO

## Regression Analysis - Underneath the hood

Need to recognise that in a sample  $\hat{\beta}$  and  $\hat{\alpha}$  are really **random variables**.  
For short EMPFT=E and WAGE\_ST=W:

$$\begin{aligned}\hat{\beta} &= \frac{\widehat{Cov}(E, W)}{\widehat{Var}(W)} \\&= \frac{\widehat{Cov}(\alpha + \beta W + u, W)}{\widehat{Var}(W)} \\&= \frac{\widehat{Cov}(\alpha, W) + \beta \widehat{Cov}(W, W) + \widehat{Cov}(u, W)}{\widehat{Var}(W)} \\&= \beta \frac{\widehat{Var}(W)}{\widehat{Var}(W)} + \frac{\widehat{Cov}(u, W)}{\widehat{Var}(W)} = \beta + \frac{\widehat{Cov}(u, W)}{\widehat{Var}(W)}\end{aligned}$$

So  $\hat{\beta}$  is a function of the random term  $u$  and hence is itself a random variable. Once  $\widehat{Cov}(E, W)$  and  $\widehat{Var}(W)$  are replaced by sample estimates we get **ONE** value which is draw from a **random distribution**.

# OLS - estimator properties

What can we learn from this?

- If  $u_{it}$  is a random variable, so is  $\hat{\beta}$
- Any particular value we get is a draw from a random distribution
- An estimator is unbiased if, on average, the estimates would be equal to the unknown  $\beta$   
at this stage the concept of unbiasedness may still be a little hazy and that is fine
- For this to happen we need to assume that  $Cov(u, x) = 0$  as then  $E(\hat{\beta}) = \beta$

Why do we need to assume this? Because while we do have values for  $x_{it}$  we do not have values for the unobserved error terms  $u_{it}$ . Hence we cannot test this. As you will find out, this is a thinking exercise and whether it is true/false/sensible/appropriate is at the core of what we do.



## OLS - the exogeneity assumption

For  $\hat{\beta}$  in  $y_{it} = \alpha + \beta x_{it} + u_{it}$  to be unbiased (i.e. on average correct) we needed

$$Cov(u_{it}, x_{it}) = 0$$

This is sometimes called the **Exogeneity assumption**. The error term has to be uncorrelated to the explanatory variable  $x_{it}$

There are a lot of reasons why this assumption may be breached.

- Simultaneity ( $WAGE\_ST \rightarrow EMPFT$  and  $EMPFT \rightarrow WAGE\_ST$ )

Discuss the fact that we have to assume that causality here goes in both directions. Hence we cannot attach one one-directional causal interpretation to the estimated coefficient. If you can estimate the model the other way round

- Omitted relevant variables or unobserved heterogeneity
- Measurement error in  $x_{it}$

## So how to make causal statements

Once we have found reasons to believe in the exogeneity assumption, the next few lectures is to introduce various standard techniques that use this assumption:

- First Difference
- Diff-in-Diff, to be used in Project
- Instrumental Variables
- Regression Discontinuity

All of them can be thought of as specific ways to apply a regression model.

# Diff-in-Diff - The Problem

Do higher minimum wages decrease employment (as predicted by a simplistic labour market model)?

# The Research Question

“This paper presents new evidence on the effect of minimum wages on establishment-level employment outcomes. We analyze the experiences of 410 fast-food restaurants in New Jersey and Pennsylvania following the increase in New Jersey’s minimum wage from \$ 4.25 to \$ 5.05 per hour. Comparisons of employment, wages, and prices at stores in New Jersey and Pennsylvania before and after the rise offer a simple method for evaluating the effects of the minimum wage.”

Card, David ; Krueger, Alan B. (1994, p.772)

## Wage distribution - Pre

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.

The easiest way to plot a histogram is

```
hist(CKdata$WAGE_ST[CKdata$STATEf == "Pennsylvania"])
```

where, in square brackets, we select that we only want data from Pennsylvania.

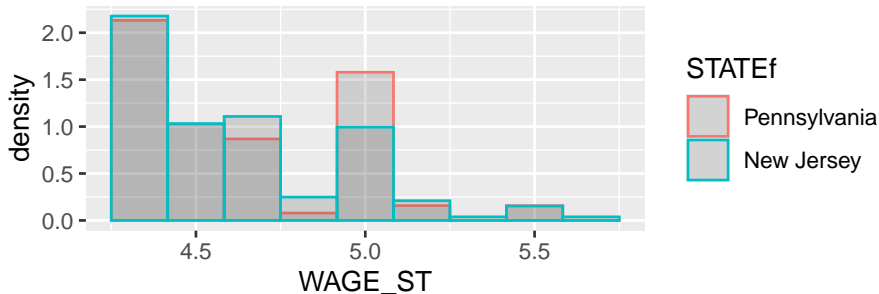
```
hist(CKdata$WAGE_ST[CKdata$STATEf == "Pennsylvania"])  
hist(CKdata$WAGE_ST[CKdata$STATEf == "New Jersey"])
```

# Wage distribution - Pre

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"))
```

Starting wage distribution, Feb/Mar 1992



## Wage distribution - Pre

Both plots show 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()
```

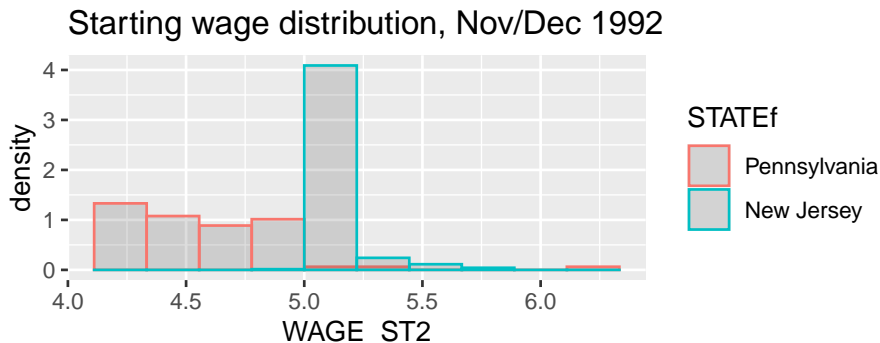
```
## # A tibble: 2 x 3  
##   STATEf      wage_FEB wage_DEC  
##   <fct>      <dbl>    <dbl>  
## 1 Pennsylvania  4.63      4.62  
## 2 New Jersey   4.61      5.08
```

Average wage in New Jersey has increased.



# Policy Evaluation - Wage distribution

```
ggplot(CKdata,aes(WAGE_ST2, colour = STATEf), colour = STATEf) +  
  geom_histogram(position="identity",  
    aes(y = ..density..),  
    bins = 10,  
    alpha = 0.2) +  
  ggtitle(paste("Starting wage distribution, Nov/Dec 1992"))
```



# Policy Evaluation - Employment outcomes

Let's measure employment before and after the policy change.

Calculate two new variables FTE and FTE2 (full time employment equivalent before and after policy change)

```
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
```

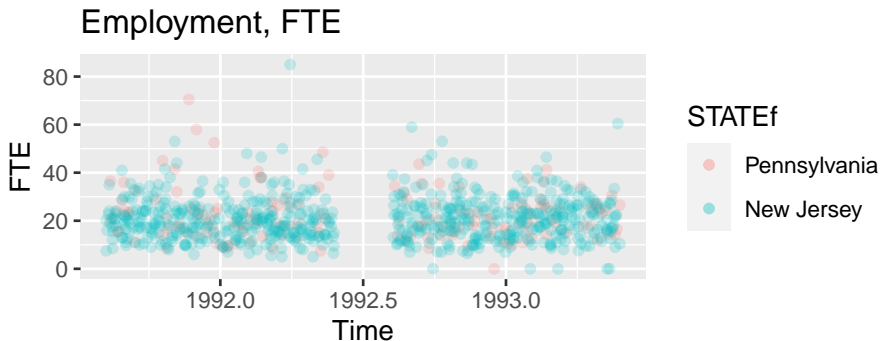
## Policy Evaluation - Diff-in-Diff estimator

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



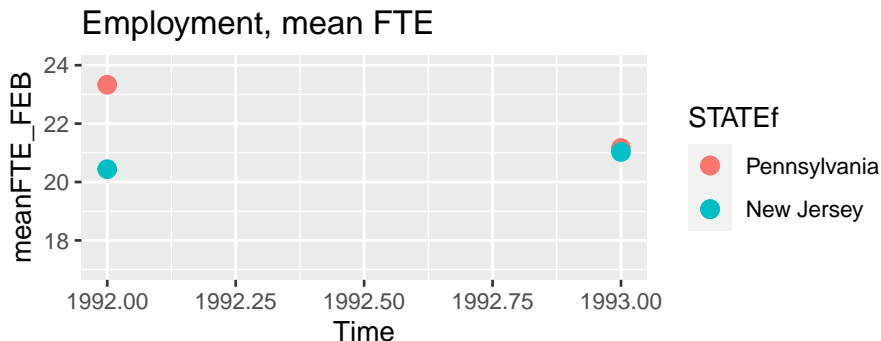
# Policy Evaluation - Diff-in-Diff estimator

```
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"))
```



# Policy Evaluation - Diff-in-Diff estimator

```
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"))
```



# Policy Evaluation - Diff-in-Diff estimator

```
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:

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

Later: This can be calculated using a regression approach (has some additional advantages)

# Outlook

Over the next weeks you will learn

- to perform more advanced statistical analysis in R, such as:
  - ▶ Hypothesis testing
  - ▶ Multivariate regression analysis
  - ▶ specification testing
- to devise methods to draw causal inference
- to understand the main pitfalls of time-series modelling and forecasting