# ANALYSIS OF PANEL DATA

Fixed-Effect and Random-Effect Models

datascience@berkeley

# Fixed-Effect Model

# An Example: The Effect of Job Training on Firm Scrap Rates

# Example 1: The Effect of Job Training on Firm Scrap Rates

- The scrap rate for a manufacturing firm is defined as the number of defective items out of every $100$ produced.
- For a given number of items produced, a decrease in scrap rate indicates a higher worker productivity.
- In this example, we use scrap rate to measure the effect of worker training on productivity.
- The data set is kindly provided by the authors of this study H. Holzer, R. Block, M. Cheatham, and J. Knott (1993), "Are Training Subsidies Effective? The Michigan Experience," Industrial and Labor Relations Review 46, 625-636
- Another influential study R.J. Lalonde (1986), "Evaluating the Econometric Evaluations of Training Programs with Experimental Data," American Economic Review 76, 604-620.
- The *jtrain2.raw* data set was provided to Professor Wooldridge by Professor Jeff Biddle at MSU, who obtained data setrom Professor Lalonde.

```
> load("jtrain.RData")
> jtrain<-data
> str(jtrain)
'data.frame':   471 obs. of  30 variables:
 $ year    : int  1987 1988 1989 1987 1988 1989 1987 1988 1989 1987 ...
 $ fcode   : num  410032 410032 410032 410440 410440 ...
 $ employ  : int  100 131 123 12 13 14 20 25 24 200 ...
 $ sales   : num  47000000 43000000 49000000 1560000 1970000 ...
 $ avgsal  : num  35000 37000 39000 10500 11000 ...
 $ scrap   : num  NA NA NA NA NA NA NA NA NA NA ...
 $ rework  : num  NA NA NA NA NA NA NA NA NA NA ...
 $ tothrs  : int  12 8 8 12 12 10 50 50 50 0 ...
 $ union   : int  0 0 0 0 0 0 0 0 0 0 ...
 $ grant   : int  0 0 0 0 0 0 0 0 0 0 ...
 $ d89     : int  0 0 1 0 0 1 0 0 1 0 ...
 $ d88     : int  0 1 0 0 1 0 0 1 0 0 ...
 $ totrain : int  100 50 50 12 13 14 15 10 20 0 ...
 $ hrsemp  : num  12 3.05 3.25 12 12 ...
 $ lscrap  : num  NA NA NA NA NA NA NA NA NA NA ...
 $ lemploy : num  4.61 4.88 4.81 2.48 2.56 ...
 $ lsales  : num  17.7 17.6 17.7 14.3 14.5 ...
 $ lrework : num  NA NA NA NA NA NA NA NA NA NA ...
 $ lhrsemp : num  2.56 1.4 1.45 2.56 2.56 ...
 $ lscrap_1: num  NA NA NA NA NA NA NA NA NA NA ...
 $ grant_1 : int  0 0 0 0 0 0 0 0 0 0 ...
 $ clscrap : num  NA NA NA NA NA NA NA NA NA NA ...
 $ cgrant  : int  0 0 0 0 0 0 0 0 0 0 ...
 $ clemploy: num  NA 0.27 -0.063 NA 0.08 ...
 $ clsales : num  NA -0.0889 0.1306 NA 0.2333 ...
 $ lavgsal : num  10.46 10.52 10.57 9.26 9.31 ...
 $ clavgsal: num  NA 0.0556 0.0526 NA 0.0465 ...
 $ cgrant_1: int  NA 0 0 NA 0 0 NA 0 0 NA ...
 $ chrsemp : num  NA -8.947 0.199 NA 0 ...
 $ clhrsemp: num  NA -1.1654 0.0478 NA 0 ...
```

```
> table(jtrain$year)

1987 1988 1989
 157  157  157
```

Showing the first 12 observations of part of the dataset

```
> head(jtrain,12)
   year  fcode employ    sales avgsal scrap rework tothrs union grant d89 d88 totrain
1  1987 410032    100 47000000  35000    NA     NA     12     0     0   0   0     100
2  1988 410032    131 43000000  37000    NA     NA      8     0     0   0   1      50
3  1989 410032    123 49000000  39000    NA     NA      8     0     0   1   0      50
4  1987 410440     12  1560000  10500    NA     NA     12     0     0   0   0      12
5  1988 410440     13  1970000  11000    NA     NA     12     0     0   0   1      13
6  1989 410440     14  2350000  11500    NA     NA     10     0     0   1   0      14
7  1987 410495     20   750000  17680    NA     NA     50     0     0   0   0      15
8  1988 410495     25   110000  18720    NA     NA     50     0     0   0   1      10
9  1989 410495     24   950000  19760    NA     NA     50     0     0   1   0      20
10 1987 410500    200 23741000  13729    NA     NA      0     0     0   0   0       0
11 1988 410500    155 19659000  14287    NA     NA      0     0     0   0   1       0
12 1989 410500     80 25992000  15758    NA     NA     24     0     0   1   0      20
```

```
      hrsemp lscrap  lemploy   lsales lrework  lhrsemp lscrap_1 grant_1 clscrap cgrant
1  12.000000      NA 4.605170 17.66566      NA 2.564949       NA       0      NA      0
2   3.053435      NA 4.875197 17.57671      NA 1.399565       NA       0      NA      0
3   3.252033      NA 4.812184 17.70733      NA 1.447397       NA       0      NA      0
4  12.000000      NA 2.484907 14.26020      NA 2.564949       NA       0      NA      0
5  12.000000      NA 2.564949 14.49354      NA 2.564949       NA       0      NA      0
6  10.000000      NA 2.639057 14.66993      NA 2.397895       NA       0      NA      0
7  37.500000      NA 2.995732 13.52783      NA 3.650658       NA       0      NA      0
8  20.000000      NA 3.218876 11.60824      NA 3.044523       NA       0      NA      0
9  41.666668      NA 3.178054 13.76422      NA 3.753418       NA       0      NA      0
10  0.000000      NA 5.298317 16.98271      NA 0.000000       NA       0      NA      0
11  0.000000      NA 5.043425 16.79405      NA 0.000000       NA       0      NA      0
12  6.000000      NA 4.382027 17.07330      NA 1.945910       NA       0      NA      0
```

List-split the data.frame using year as the factor.

```
X<-split.data.frame(jtrain, as.factor(jtrain$year))
str(X)

jtrain.87 <- X$`1987`
jtrain.88 <- X$`1988`
jtrain.89 <- X$`1989`
str(jtrain.87)
```

```
List of 3
 $ 1987:'data.frame':   157 obs. of  30 variables:
  ..$ year    : int [1:157] 1987 1987 1987 1987 1987 1987 1987 1987 1987 1987 ...
  ..$ fcode   : num [1:157] 410032 410440 410495 410500 410501 ...
  ..$ employ  : int [1:157] 100 12 20 200 NA NA 15 24 48 17 ...
  ..$ sales   : num [1:157] 47000000 1560000 750000 23741000 6000000 ...
  ..$ avgsal  : num [1:157] 35000 10500 17680 13729 NA ...
  ..$ scrap   : num [1:157] NA NA NA NA NA NA NA NA NA NA ...
  ..$ rework  : num [1:157] NA NA NA NA NA NA NA NA NA NA ...
  ..$ tothrs  : int [1:157] 12 12 50 0 0 0 0 0 14 150 ...
  ..$ union   : int [1:157] 0 0 0 0 0 0 0 1 0 0 ...
  ..$ grant   : int [1:157] 0 0 0 0 0 0 0 0 0 0 ...
  ..$ d89     : int [1:157] 0 0 0 0 0 0 0 0 0 0 ...
  ..$ d88     : int [1:157] 0 0 0 0 0 0 0 0 0 0 ...
  ..$ totrain : int [1:157] 100 12 15 0 10 0 0 0 3 5 ...
  ..$ hrsemp  : num [1:157] 12 12 37.5 0 NA ...
  ..$ lscrap  : num [1:157] NA NA NA NA NA NA NA NA NA NA ...
  ..$ lemploy : num [1:157] 4.61 2.48 3 5.3 NA ...
  ..$ lsales  : num [1:157] 17.7 14.3 13.5 17 15.6 ...
  ..$ lrework : num [1:157] NA NA NA NA NA NA NA NA NA NA ...
  ..$ lhrsemp : num [1:157] 2.56 2.56 3.65 0 NA ...
  ..$ lscrap_1: num [1:157] NA NA NA NA NA NA NA NA NA NA ...
  ..$ grant_1 : int [1:157] 0 0 0 0 0 0 0 0 0 0 ...
  ..$ clscrap : num [1:157] NA NA NA NA NA NA NA NA NA NA ...
  ..$ cgrant  : int [1:157] 0 0 0 0 0 0 0 0 0 0 ...
  ..$ clemploy: num [1:157] NA NA NA NA NA NA NA NA NA NA ...
  ..$ clsales : num [1:157] NA NA NA NA NA NA NA NA NA NA ...
  ..$ lavgsal : num [1:157] 10.46 9.26 9.78 9.53 NA ...
  ..$ clavgsal: num [1:157] NA NA NA NA NA NA NA NA NA NA ...
  ..$ cgrant_1: int [1:157] NA NA NA NA NA NA NA NA NA NA ...
  ..$ chrsemp : num [1:157] NA NA NA NA NA NA NA NA NA NA ...
  ..$ clhrsemp: num [1:157] NA NA NA NA NA NA NA NA NA NA ...
```

```
> str(jtrain.87)
'data.frame':   157 obs. of  30 variables:
 $ year    : int  1987 1987 1987 1987 1987 1987 1987 1987 1987 1987 ...
 $ fcode   : num  410032 410440 410495 410500 410501 ...
 $ employ  : int  100 12 20 200 NA NA 15 24 48 17 ...
 $ sales   : num  47000000 1560000 750000 23741000 6000000 ...
 $ avgsal  : num  35000 10500 17680 13729 NA ...
 $ scrap   : num  NA NA NA NA NA NA NA NA NA NA ...
 $ rework  : num  NA NA NA NA NA NA NA NA NA NA ...
 $ tothrs  : int  12 12 50 0 0 0 0 0 14 150 ...
 $ union   : int  0 0 0 0 0 0 0 1 0 0 ...
 $ grant   : int  0 0 0 0 0 0 0 0 0 0 ...
 $ d89     : int  0 0 0 0 0 0 0 0 0 0 ...
 $ d88     : int  0 0 0 0 0 0 0 0 0 0 ...
 $ totrain : int  100 12 15 0 10 0 0 0 3 5 ...
 $ hrsemp  : num  12 12 37.5 0 NA ...
 $ lscrap  : num  NA NA NA NA NA NA NA NA NA NA ...
 $ lemploy : num  4.61 2.48 3 5.3 NA ...
 $ lsales  : num  17.7 14.3 13.5 17 15.6 ...
 $ lrework : num  NA NA NA NA NA NA NA NA NA NA ...
 $ lhrsemp : num  2.56 2.56 3.65 0 NA ...
 $ lscrap_1: num  NA NA NA NA NA NA NA NA NA NA ...
 $ grant_1 : int  0 0 0 0 0 0 0 0 0 0 ...
 $ clscrap : num  NA NA NA NA NA NA NA NA NA NA ...
 $ cgrant  : int  0 0 0 0 0 0 0 0 0 0 ...
 $ clemploy: num  NA NA NA NA NA NA NA NA NA NA ...
 $ clsales : num  NA NA NA NA NA NA NA NA NA NA ...
 $ lavgsal : num  10.46 9.26 9.78 9.53 NA ...
 $ clavgsal: num  NA NA NA NA NA NA NA NA NA NA ...
 $ cgrant_1: int  NA NA NA NA NA NA NA NA NA NA ...
 $ chrsemp : num  NA NA NA NA NA NA NA NA NA NA ...
 $ clhrsemp: num  NA NA NA NA NA NA NA NA NA NA ...
```

This dataset has only 157 observations and 30 variables, containing only the cross-section units in ~~187~~.

157

- Let's for the time being not take advantage of information provided by multiple panels of cross-sectional units and estimate a model using only information in 1987.

- Specifically, we will examine the relationship between scrap rate and training, conditional on firm size (measured in sales and number of employees)

```
summary(cbind(jtrain.87$lscrap,jtrain.87$hrsemp,jtrain.87$lsales,jtrain.87$lem
ploy))
```

```
       V1                 V2                 V3                 V4
 Min.   :-4.6052   Min.   :  0.000   Min.   :12.64   Min.   :1.386
 1st Qu.: 0.0000   1st Qu.:  0.000   1st Qu.:14.18   1st Qu.:2.708
 Median : 0.5158   Median :  0.000   Median :14.88   Median :3.277
 Mean   : 0.5974   Mean   :  8.887   Mean   :14.92   Mean   :3.449
 3rd Qu.: 1.7918   3rd Qu.: 10.000   3rd Qu.:15.74   3rd Qu.:4.248
 Max.   : 3.4012   Max.   :100.000   Max.   :17.67   Max.   :6.184
 NA's   :103       NA's   :28        NA's   :38      NA's   :13
```

```
# hrsemp: annual hours of training per employee
# sales : annual firm sales in dollar
# employ: number of firm employee
```

```
> summary(jtrain.87$scrap)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
  0.010   1.000   1.675   4.612   6.000  30.000     103
> summary(jtrain.87$sales)
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.     NA's
  307500  1433000  2900000  5341000  6885000 47000000       38
```

```
jtrain.87.ols <- lm(lscrap ~ hrsemp+lsales+lemploy,data=jtrain.87)
summary(jtrain.87.ols)
```

```
Call:
lm(formula = lscrap ~ hrsemp + lsales + lemploy, data = jtrain.87)

Residuals:
     Min       1Q   Median       3Q      Max
-2.81878 -0.91530  0.03304  0.87052  2.68042

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 11.74426    4.57470   2.567  0.01420 *
hrsemp      -0.04218    0.01868  -2.259  0.02957 *
lsales      -0.95064    0.36984  -2.570  0.01409 *
lemploy      0.99213    0.35692   2.780  0.00833 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.3 on 39 degrees of freedom
  (114 observations deleted due to missingness)
Multiple R-squared:  0.3099,    Adjusted R-squared:  0.2568
F-statistic: 5.838 on 3 and 39 DF,  p-value: 0.002148
```

Berkeley SCHOOL OF INFORMATION