ANALYSIS OF PANEL DATA

Fixed-Effect and Random-Effect Models

datascience@berkeley

An Introduction to Fixed-Effect Models

Fixed-Effect Transformation

- Recall from the last lecture that we consider the following models:

$$y_{it} = \beta_0 + \beta_1 x_{it} + a_i + \epsilon_{it}$$

where $i = 1, 2, \ldots, n$ and $t = 1, 2, \ldots, T$

- An alternative way to eliminate the time-invariant unobserved variable is the **fixed effect transformation**.

- Fixed effect transformation uses the average of individual over time and the "**average equation**" from the original equation. Averaging individuals over time, we get

$$\bar{y}_i = \beta_0 + \beta_1 \bar{x}_i + \bar{a}_i + \bar{\epsilon}_i$$

- Fixed effect transformation uses the average of individual over time and the "**average equation**" from the original equation. Averaging individuals over time, we get

$$\bar{y}_i = \beta_0 + \beta_1 \bar{x}_i + \bar{a}_i + \bar{\epsilon}_i$$

Substracting it from the original model, we obtain

$$\left(y_{it} - \bar{y}_i\right) = \beta_1 \left(x_{it} - \bar{x}_i\right) + \left(\epsilon_{it} - \bar{\epsilon}_i\right)$$

where $\bar{y}_i = \frac{1}{n} \sum_{t=1}^{T} y_{it}$ and $\bar{x}_i$ is defined similarly. The model can be expressed more compactly in the **time-demeaned** form:

$$\left(y_{it} - \bar{y}_i\right) = \beta_1 \left(x_{it} - \bar{x}_i\right) + \left(\epsilon_{it} - \bar{\epsilon}_i\right)$$

where $y_{it} - \bar{y}_i$ is the time demeanded dependent (or response) variable.

A More General Form

Fixed-effect, potentially correlated with explanatory variables

$$y_{it} = \beta_1 x_{it1} + \cdots + \beta_k x_{itk} + a_i + u_{it}, \; i = 1, \ldots, N, t = 1, \ldots, T$$

Form time averages for each individual.

$$\bar{y}_i = \beta_1 \bar{x}_{i1} + \cdots + \beta_k \bar{x}_{ik} + \bar{a}_i + \bar{u}_i$$

$$\Rightarrow [y_{it} - \bar{y}_i] = \beta_1 [x_{it1} - \bar{x}_{i1}] + \cdots + \beta_k [x_{itk} - \bar{x}_{ik}] + [u_{it} - \bar{u}_i]$$

Because $a_i - \bar{a}_i = 0$ (the fixed effect is removed)

- **Estimate time-demeaned equation by OLS.**
  - Uses time variation within cross-sectional units (= within-estimator).

Within Estimation

- The *fixed effect transformation* is also called *within transformation*, in which *within* can be read as within each of the subjects in the dataset: each of the cross-sectional subjects' data is demeaned leveraging on time variation in both y and x and more importantly, the *unobserved individual heterogeneity* is eliminated within each of the individuals.

- The *fixed effect estimator* is also called *within estimator*.

- Because the transformation relies on time variation within each of the cross-sectional subjects, those variables without much variation to begin with will become (almost) a constant after the transformation, resulting in imprecise estimates.

Between Estimation

- **Between Estimators:** The OLS estimators on the cross-sectional average equation introduced above, from which we substract the cross-sectional equation, is called the $.

- We will not discuss the $, as it is biased when the observed explanatory variables, $x's$, are correlated with the unobserved fixed effect, a_i. Also, it does not utilize the panel data efficiently: it does not leverage on the time varying information.

- If we think that the observed explanatory variables and the unobserved fixed effect are not correlated, then we should us **random effect model**, which we will discuss later in this lecture.

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
```

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

The plm Package

```
plm(formula, data, subset, na.action, effect = c("individual", "time", "twoways"),
    model = c("within", "random", "ht", "between", "pooling", "fd"),
    random.method = c("swar", "walhus", "amemiya", "nerlove", "kinla"),
    random.dfcor = NULL,
    inst.method = c("bvk", "baltagi", "am", "bmc"), restrict.matrix = NULL,
    restrict.rhs = NULL, index = NULL, ...)
```

plm is a general function for the estimation of linear panel models. It supports the following estimation methods: pooled OLS (model = "pooling"), fixed effects ("within"), random effects ("random"), first–differences ("fd"), and between ("between"). It supports unbalanced panels and two–way effects (although not with all methods).

For random effects models, four estimators of the transformation parameter are available by setting random.method to one of "swar" (Swamy and Arora (1972)) (default), "amemiya" (Amemiya (1971)), "walhus" (Wallace and Hussain (1969)), or "nerlove" (Nerlove (1971)).

Instrumental variables estimation is obtained using two–part formulas, the second part indicating the instrumental variables used. This can be a complete list of instrumental variables or an update of the first part. If, for example, the model is y ~ x1 + x2 + x3, with x1 and x2 endogenous and z1 and z2 external instruments, the model can be estimated with:

- formula=y~x1+x2+x3 | x3+z1+z2,
- formula=y~x1+x2+x3 | .-x1-x2+z1+z2.

- To structure a panel data, we use each row of the data for a specific individual in a particular time period.
- In $ data.frame can be used, but it includes an argument called *index* to indicate the structure of the data. This can be:
- NULL (the default value), it is then assumed that the first two columns contain the individual and the time index and that observations are ordered by individual and by time period
- a character string, which should be the name of the individual index
- a character vector of length two containing the names of the individual and the time index
- an integer which is the number of individuals (only in case of a balanced panel with observations ordered by individual)

- The pdata.frame function is then called internally, which returns a pdata.frame which is a data.frame with an attribute called index.
- The **plm** package is very rich and provides four estimation function. In this course, we will only use the *plm* function, as it already provides estimation for both fixed effect and random effect models.
- A nice feature of these functions is that is share the same interface as that of the *lm()* function. Their first two arguments are **formula** and **data**
- Of the other arguments, I want to highlight **index**, which we discussed above, and **$**, which is use to indicate the kind of effects to be included in the model. That is, *individual effect, time effect*, or both.
- Data Transformation: The *within* transformation, $Q = I_{nT} - P$, where $P = \frac{1}{T} I_n xjj'$ returns a vector containing individual means and I_z is a $z \times z$ identity matrix.
- The **within** function can be used to perform the within transformation

Value

An object of class `c("plm","panelmodel")`.

A `"plm"` object has the following elements :

`coefficients`	the vector of coefficients,
`vcov`	the covariance matrix of the coefficients,
`residuals`	the vector of residuals,
`df.residual`	degrees of freedom of the residuals,
`formula`	an object of class `'pFormula'` describing the model,
`model`	a data.frame of class `'pdata.frame'` containing the variables used for the estimation: the response is in first column and the two indexes in the two last columns,
`ercomp`	an object of class `'ercomp'` providing the estimation of the components of the errors (for random effects models only),
`call`	the call.

Effect of Job Training Program on Scrap Rates Revisit

```
# Use the plm.data() function to transform a data frame in a format suitable
for using with the estimation functions of plm.
jtrain.panel<-plm.data(jtrain, c("fcode","year"))
summary(jtrain.panel)
```

Description

This function transforms a data frame in a format suitable for using with the estimation functions of plm.

Usage

```
plm.data(x, indexes = NULL)
```

Arguments

x	a data.frame,
indexes	a vector (of length one or two) indicating the (individual and time) indexes.

indexes can be:

- a character string which is the name of the individual index variable, in this case a new variable called "time" containing the time index is added,
- an integer, the number of individuals in the case of balanced panel, in this case two new variables "time" and "id" containing the individual and the time indexes are added,
- a vector of two character strings which contains the names of the individual and of the time indexes.

```
> summary(jtrain.panel)
     fcode          year          employ            sales             avgsal
 410032 :  3   1987:157   Min.   :  4.00   Min.   :  110000   Min.   : 4237
 410440 :  3   1988:157   1st Qu.: 15.00   1st Qu.: 1550000   1st Qu.:14102
 410495 :  3   1989:157   Median : 30.00   Median : 3000000   Median :17773
 410500 :  3              Mean   : 59.32   Mean   : 6116037   Mean   :18873
 410501 :  3              3rd Qu.: 72.00   3rd Qu.: 7700000   3rd Qu.:22360
 410509 :  3              Max.   :525.00   Max.   :54000000   Max.   :42583
 (Other):453              NA's   :31       NA's   :98         NA's   :65
     scrap            rework           tothrs           union            grant
Min.   : 0.0100   Min.   : 0.000   Min.   :  0.0   Min.   :0.0000   Min.   :0.0000
1st Qu.: 0.5925   1st Qu.: 0.350   1st Qu.:  0.0   1st Qu.:0.0000   1st Qu.:0.0000
Median : 1.4150   Median : 1.160   Median : 12.0   Median :0.0000   Median :0.0000
Mean   : 3.8436   Mean   : 3.474   Mean   : 29.2   Mean   :0.1975   Mean   :0.1401
3rd Qu.: 4.0000   3rd Qu.: 4.000   3rd Qu.: 40.0   3rd Qu.:0.0000   3rd Qu.:0.0000
Max.   :30.0000   Max.   :40.000   Max.   :320.0   Max.   :1.0000   Max.   :1.0000
NA's   :309       NA's   :348      NA's   :56
      d89              d88             totrain          hrsemp           lscrap
Min.   :0.0000   Min.   :0.0000   Min.   :  0.00   Min.   :  0.000   Min.   :-4.6052
1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:  0.00   1st Qu.:  0.000   1st Qu.:-0.5234
Median :0.0000   Median :0.0000   Median :  8.00   Median :  3.308   Median : 0.3471
Mean   :0.3333   Mean   :0.3333   Mean   : 23.09   Mean   : 14.968   Mean   : 0.3937
3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.: 25.00   3rd Qu.: 18.663   3rd Qu.: 1.3863
Max.   :1.0000   Max.   :1.0000   Max.   :350.00   Max.   :163.917   Max.   : 3.4012
                                  NA's   :6        NA's   :81        NA's   :309
    lemploy          lsales          lrework          lhrsemp          lscrap_1
Min.   :1.386   Min.   :11.61   Min.   :-4.6052   Min.   :0.000   Min.   :-4.6052
1st Qu.:2.708   1st Qu.:14.25   1st Qu.:-0.9163   1st Qu.:0.000   1st Qu.:-0.2675
Median :3.401   Median :14.91   Median : 0.1823   Median :1.460   Median : 0.4414
Mean   :3.531   Mean   :15.03   Mean   : 0.1642   Mean   :1.650   Mean   : 0.5129
3rd Qu.:4.277   3rd Qu.:15.86   3rd Qu.: 1.3863   3rd Qu.:2.979   3rd Qu.: 1.6094
Max.   :6.263   Max.   :17.80   Max.   : 3.6889   Max.   :5.105   Max.   : 3.4012
NA's   :31      NA's   :98      NA's   :350       NA's   :81      NA's   :363
```

First-Difference Method

```
> jtrain.fd <- plm(lscrap ~ hrsemp+lsales+lemploy,data=jtrain.panel, model="fd")
> summary(jtrain.fd)
Oneway (individual) effect First-Difference Model

Call:
plm(formula = lscrap ~ hrsemp + lsales + lemploy, data = jtrain.panel,
    model = "fd")

Unbalanced Panel: n=47, T=1-3, N=135

Residuals :
   Min. 1st Qu.  Median 3rd Qu.    Max.
-3.0600 -0.1110  0.0838  0.2770  2.6500

Coefficients :
             Estimate Std. Error t-value Pr(>|t|)
(intercept) -0.2112469  0.0734771 -2.8750 0.005118 **
hrsemp      -0.0012945  0.0022429 -0.5771 0.565393
lsales      -0.3475582  0.3436699 -1.0113 0.314770
lemploy      0.2589719  0.4105471  0.6308 0.529886
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Total Sum of Squares:    34.021
Residual Sum of Squares: 33.353
R-Squared:      0.019647
Adj. R-Squared: 0.018754
F-statistic: 0.561147 on 3 and 84 DF, p-value: 0.64214
```

Fixed-Effect Method

```
> jtrain.fe2 <- plm(lscrap ~ hrsemp+lsales+lemploy,data=jtrain.panel, model="within")
> summary(jtrain.fe2)
Oneway (individual) effect Within Model

Call:
plm(formula = lscrap ~ hrsemp + lsales + lemploy, data = jtrain.panel,
    model = "within")

Unbalanced Panel: n=47, T=1-3, N=135

Residuals :
     Min.   1st Qu.    Median   3rd Qu.      Max.
-1.800000 -0.124000  0.000966  0.136000  1.610000

Coefficients :
          Estimate Std. Error t-value Pr(>|t|)
hrsemp  -0.0041585  0.0024653 -1.6868  0.09532 .
lsales  -0.5616153  0.3611430 -1.5551  0.12364
lemploy  0.3655257  0.4389659  0.8327  0.40735
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Total Sum of Squares:    27.287
Residual Sum of Squares: 24.889
R-Squared:      0.087879
Adj. R-Squared: 0.055331
F-statistic: 2.7298 on 3 and 85 DF, p-value: 0.048909
```

A Digression: Differencing When There Are More Than Two Time Periods

Differencing with More than Two Time Periods

- Suppose we have N individuals and 3 time periods for each individual, totalling $3N$ observations. A general fixed effect model can be written as

$$y_{it} = \delta_1 + \delta_2 d2_t + \delta_3 d3_t + \beta_1 x_{it1} + \cdots + \beta_k x_{itk} + a_i + \epsilon_{it}$$

for $t = 1, 2, 3$

- The key assumption is that the error terms are uncorrelated with the explanatory variable in each time period:

$$Cov(x_{itj}, \epsilon_{is})$$

for all t, s, j

- This means that the explanatory variables are *strictly exogenous* after the unobserved effect a_i is eliminated.

- If an important time-varying variable is omitted from the model, then this assumption is violated.

Pause and Think (2 minutes):

Let's spend a couple of minutes to think about this assumption. It would help to write out the time index.

- The key assumption is that the error terms are uncorrelated with the explanatory variable in each time period:

$$Cov(x_{itj}, \epsilon_{is})$$

for all t, s, j

- This means that the explanatory variables are *strictly exogenous* after the unobserved effect a_i is eliminated.

- If a_i is correlated with x_{itj}, then x_{itj} will be correlated with the composite error, $v_{it} = a_i + \epsilon_{it}$.
- However, we can eliminate a_i by differencing adjacent periods.
- In the case where $T = 3$, we can substract time period one from time period two, time period two from time period three.

$$\triangle y_{it} = \delta_2 \triangle d2_t + \delta_3 \triangle d3_t + \beta_1 \triangle x_{it1} + \cdots + \triangle x_{itk} + \triangle \epsilon_{it}$$

for $t = 1, 2$

- If the equation satisfies the classical linear model assumptions, then pooled OLS gives unbiased estimators, and t and F statistics are valid for hypothesis testing. Asymptotic results can be used as well.
- As long as $\triangle \epsilon_{it}$ is uncorrelated with $\triangle x_{itj}$ for all j and $t = 2, 3$, then the OLS estimators are also consistent.

Remarks on Fixed-Effect Models

Deterministic Time-Varying and Time-Invariant Variables

- The fixed effect estimator allows for correlation between a_i and the explanatory variables in any time period, as in the case in first-differencing.
- A side-effect is that all of the time-invariant variables are eliminated alongisde with the unobserved fixed effect. As such, variables such as gender, credit score at loan origination, a biometric measure at the beginning of using a wearable, the distance between the center of a city to the nearest port, etc., will all be swept away by the fixed effect transformation. As such, the effect of time-invariant variables cannot be estimated
- That said, the effect of interactions with time-invariant variables can be estimated (e.g. the interaction of education with time dummies)
- If a full set of time dummies are included, the effect of deterministic time-varying variables (e.g. experience) cannot be estimated because they cannot be distinguished from the aggregate time effect.

Assumptions Required for valid OLS Estimation

- Under a $ assumption on the explanatory variables, the fixed effect estimator is unbiased: the error term ϵ_{it} is uncorrelated with all of the explanatory variables across all time period:

$$E(\epsilon_{it}|X_i, a_i) = 0 \forall t$$

- Other assumptions require that the error term be homoskedastic and serially uncorrelated across t.

R^2 and Degree of Freedom in a Fixed Effect Model

- Note that the R-squared of the fixed effect equation (i.e. demeaned equation) should be interpreted with caution.
- It measures the amount of time variation in y_{it} that can be explained by the variation of the explanatory variables.
- In a general fixed effect model, we have $N \times T$ observations and k independent variables. As such, we should have $NT - k$ degree of freedom. Is that correct?

Berkeley SCHOOL OF INFORMATION

ANALYSIS OF PANEL DATA

Fixed-Effect and Random-Effect Models

datascience@berkeley

- This is incorrect, because for each of the cross-sectional unit i, we lose one df when doing the time-demeaning: for each i, the demeaned errors ϵ_{it} add up to zero when summed across t, so one df is lost. This is taken care off in modern regression packages that have fixed effect estimation function. If, however, we do the time-demeaning and then pooled OLS manually, we will have to adjust the degree of freedom.

Consider one of the datasets that come with the plm package.

```
library(plm)
data("Grunfeld", package="plm")
head(Grunfeld)
```

	firm	year	inv	value	capital
1	1	1935	317.6	3078.5	2.8
2	1	1936	391.8	4661.7	52.6
3	1	1937	410.6	5387.1	156.9
4	1	1938	257.7	2792.2	209.2
5	1	1939	330.8	4313.2	203.4
6	1	1940	461.2	4643.9	207.2
7	1	1941	512.0	4551.2	255.2
8	1	1942	448.0	3244.1	303.7
9	1	1943	499.6	4053.7	264.1
10	1	1944	547.5	4379.3	201.6
11	1	1945	561.2	4840.9	265.0
12	1	1946	688.1	4900.9	402.2
13	1	1947	568.9	3526.5	761.5
14	1	1948	529.2	3254.7	922.4
15	1	1949	555.1	3700.2	1020.1
16	1	1950	642.9	3755.6	1099.0
17	1	1951	755.9	4833.0	1207.7
18	1	1952	891.2	4924.9	1430.5
19	1	1953	1304.4	6241.7	1777.3
20	1	1954	1486.7	5593.6	2226.3
21	2	1935	209.9	1362.4	53.8
22	2	1936	355.3	1807.1	50.5
23	2	1937	469.9	2676.3	118.1
24	2	1938	262.3	1801.9	260.2
25	2	1939	230.4	1957.3	312.7

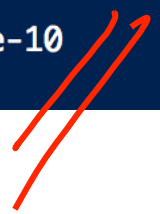
```
'data.frame':   471 obs. of  30 variables:
 $ year    : int  1987 1988 1989 1987 1988 1989 1987 1988 1989 1987 ...
 $ fcode   : num  410032 410032 410032 410440 410440 ...
 $ employ  : int  100 131 123 12 13 14 20 25 24 200 ...
 $ sales   : num  47000000 43000000 49000000 1560000 1970000 ...
 $ avgsal  : 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 ...
```

Tests of Poolability

```
> znp <- pvcm(inv~value+capital,data=Grunfeld, model="within")
> zplm<- plm(inv~value+capital,data=Grunfeld)
> pooltest(zplm,znp)

        F statistic

data:  inv ~ value + capital
F = 5.7805, df1 = 18, df2 = 170, p-value = 1.219e-10
alternative hypothesis: unstability
```



```
> pooltest(inv~value+capital, data=Grunfeld, model="within")

        F statistic

data:  inv ~ value + capital
F = 5.7805, df1 = 18, df2 = 170, p-value = 1.219e-10
alternative hypothesis: unstability
```

Test for Serial Correlation

```
# Breusch-Godfrey and Durbin-Watson Test
# This test shares their OLS counterparts and allows for higher-order serial
correlation

# As a function, it is simply a wrapper of the bgtest and dwtest. So, all the
arguments from these two tests apply and may be passed on through `...`
operator.

pbgtest(grun.fe,order=2)
```

```
> pbgtest(grun.fe,order=2)

        Breusch-Godfrey/Wooldridge test for serial correlation in panel models

data:  inv ~ value + capital
chisq = 42.587, df = 2, p-value = 5.655e-10
alternative hypothesis: serial correlation in idiosyncratic errors
```

Random-Effect Models

Random Effect Models

Recall the general linear regression models with unobserved individual effect:

$$y_{it} = \beta_0 + \beta_1 x_{1it} + \cdots + \beta_k x_{kit} + a_i + \epsilon_{it}$$

where $i = 1, 2, \ldots, n$ and $t = 1, 2, \ldots, T$

- If the unobserved individual effect a_i is uncorrelated with the explanatory variables, the techniques described above are not needed to produce a consistent estimator.

$$Cov(x_{itj}, a_i) = 0$$

for $t = 1, 2, \ldots, T$ and $j = 1, 2, \ldots, k$

- What is means is that the random effect assumptions include all of the fixed effect assumptions plus the additional (strong) requirement that a_i is independent of all explanatory variables in all time periods in the model.

- How should we estimate β_j in the above unobserved effect model?
- Note that under these assumptions, we can use the *random effect* models to obtain consistent OLS estimators using only a single cross section: there is no need to the panel data at all if the objective is to obtain consistent estimators for β_j.
- Of course, using a single cross section means we throw away potentially valuable information offered by panel data.
- Let's rewrite the model using a composite error term:

$$y_{it} = \beta_0 + \beta_1 x_{1it} + \cdots + \beta_k x_{kit} + \nu_{it}$$

where $\nu_{it} = a_i + \epsilon_{it}$ $i = 1, 2, \ldots, n$ and $t = 1, 2, \ldots, T$

- Because a_i is contained in the composite error term in each time period, ν_{it} is serially correlated. Under Random Effect assumptions, we have

$$Corr(\nu_{it}, \nu_{is}) = \frac{\sigma_a^2}{\sigma_a^2 + \sigma_\epsilon^2}$$

- In other words, when $E(X_{it} a_i) \neq 0$, panel data provides a valuable tool for eliminating omitted variables bias. We use Fixed Effects to gain the benefits of panel data.

- When $E(X_{it} a_i) = 0$, panel data does not offer special benefits. We use Random Effects to overcome the serial correlation of panel data.

- The correlation in the error term can be substantial. Because pooled OLS standard errors ignore this correlation, they will be in correct, as will the usual test statistics.

- A solution to this problem is to use generalized least square (GLS).

- For this procedure to come with good properties, we need large N and small T. This is, a short panel.

- Deriving the GLS transformation to eliminate serial correlation requires quite a bit of matrix algebra. However, the transformation itself is pretty simple:

$$\lambda = 1 - \left[\frac{\sigma_\epsilon^2}{\sigma_\epsilon^2 + T\sigma_a^2} \right]^{1/2}$$

which falls between 0 and 1.

- The transformed model becomes

$$y_{it} - \lambda \bar{y}_i = \beta_0(1-\lambda) + \beta_1 \left(x_{1it} - \lambda \bar{x}_{1i}\right) + \cdots + \beta_k \left(x_{kit} - \lambda \bar{x}_{ki}\right) + \left(\nu_{it} - \lambda \bar{\nu}_i\right)$$

where the *overbar* denotes the time averages.

- While the FE estimators subtracts the time averages from the corresponding variable, the random effect transformation subtracts a fraction of that time average with the fraction being a function of σ_ϵ^2, σ_a^2 and T.

- The GLS estimator is simply the pooled estimator of the above model.

- One of the advantage of random effect model (relative to fixed effect model) is that it allows for time invariant explanatory variables to be included in the model; a fixed effect models eliminates all the time invariant (observed and unobserved) variables.

- In practice, λ needs to be estimated:

$$1 - \left[\frac{1}{1 + T(\hat{\sigma}_a^2 / \hat{\sigma}_\epsilon^2)} \right]^{1/2}$$

where $\hat{\sigma}_a^2$ and $\hat{\sigma}_\epsilon^2$ are consistent estimators of σ_a^2 and σ_ϵ^2.

- These estimators can be based on pooled OLS or fixed effect residuals.
- In practice, random effect models can be implemented easily by modern econometric packages, such as *plm*, and λ can be automated computed as well.
- The *feasible* GLS estimator that uses $\hat{\lambda}$ in place of λ is called the **random effect estimator**.
- Under the random effect assumptions, the estimator is *consistent* and *asymptotically normally distributed* for large N and fixed T.
- In applications of FE and RE, it is usually informative also to compute the pooled OLS estimates as well because comparing the three sets of estimates can help determine the nature of the biases caused by leaving the unobserved effect, a_i, entirely in the error term (as does pooled OLS) or partially in the error term (as does the RE transformation).

- But we must remember that, even if ai is uncorrelated with all explanatory variables in all time periods, the pooled OLS standard errors and test statistics are generally invalid: they ignore the often substantial serial correlation in the composite errors, $\nu_{it} = a_i + \epsilon_{it}$

Example: A Wage Equation Using Panel Data

- Let's use use the data in *wagepan.RData* to estimate a wage equation for men.

- Specifically, we use three methods: pooled OLS, random effects, and fixed effects. The first two methods include *educ* and *race dummies (black and hispan)*, but these drop out of the fixed effects model. The time- varying variables are *exper, exper2, union*, and *married*.

Three Different Estimators of a Wage Equation

Dependent Variable: log(*wage*)			
Independent Variables	**Pooled OLS**	**Random Effects**	**Fixed Effects**
educ	.091 (.005)	.092 (.011)	———
black	−.139 (.024)	−.139 (.048)	———
hispan	.016 (.021)	.022 (.043)	———
exper	.067 (.014)	.106 (.015)	———
$exper^2$	−.0024 (.0008)	−.0047 (.0007)	−.0052 (.0007)
married	.108 (.016)	.064 (.017)	.047 (.018)
union	.182 (.017)	.106 (.018)	.080 (.019)

- The coefficients on educ, black, and hispan are similar for the pooled OLS and random effects ~~esti- mations~~.
- The pooled OLS standard errors are the usual OLS standard errors, and these underestimate the true standard errors because they ignore the positive serial correlation, but we report them here for comparison only.
- The experience profile is somewhat different, and both the marriage and union *premiums* fall notably in the random effects estimation.
- Note that when we eliminate the unobserved effect entirely by using fixed effects, the marriage premium falls to about 4.7, although it is still statistically significant.

A partial list of the dataset

```
'data.frame':   4360 obs. of  44 variables:
 $ nr      : int  13 13 13 13 13 13 13 13 17 17 ...
 $ year    : int  1980 1981 1982 1983 1984 1985 1986 1987 1980 1981 ...
 $ agric   : int  0 0 0 0 0 0 0 0 0 0 ...
 $ black   : int  0 0 0 0 0 0 0 0 0 0 ...
 $ bus     : int  1 0 1 1 0 1 1 1 0 0 ...
 $ construc: int  0 0 0 0 0 0 0 0 0 0 ...
 $ ent     : int  0 0 0 0 0 0 0 0 0 0 ...
 $ exper   : int  1 2 3 4 5 6 7 8 4 5 ...
 $ fin     : int  0 0 0 0 0 0 0 0 0 0 ...
 $ hisp    : int  0 0 0 0 0 0 0 0 0 0 ...
 $ poorhlth: int  0 0 0 0 0 0 0 0 0 0 ...
 $ hours   : int  2672 2320 2940 2960 3071 2864 2994 2640 2484 2804 ...
 $ manuf   : int  0 0 0 0 0 0 0 0 0 0 ...
 $ married : int  0 0 0 0 0 0 0 0 0 0 ...
 $ min     : int  0 0 0 0 0 0 0 0 0 0 ...
 $ nrthcen : int  0 0 0 0 0 0 0 0 0 0 ...
 $ nrtheast: int  1 1 1 1 1 1 1 1 1 1 ...
 $ occ1    : int  0 0 0 0 0 0 0 0 0 0 ...
 $ occ2    : int  0 0 0 0 0 1 1 1 1 1 ...
 $ occ3    : int  0 0 0 0 0 0 0 0 0 0 ...
 $ occ4    : int  0 0 0 0 0 0 0 0 0 0 ...
 $ occ5    : int  0 0 0 0 1 0 0 0 0 0 ...
 $ occ6    : int  0 0 0 0 0 0 0 0 0 0 ...
 $ occ7    : int  0 0 0 0 0 0 0 0 0 0 ...
 $ occ8    : int  0 0 0 0 0 0 0 0 0 0 ...
 $ occ9    : int  1 1 1 1 0 0 0 0 0 0 ...
 $ per     : int  0 1 0 0 1 0 0 0 0 0 ...
 $ pro     : int  0 0 0 0 0 0 0 0 0 0 ...
 $ pub     : int  0 0 0 0 0 0 0 0 0 0 ...
 $ rur     : int  0 0 0 0 0 0 0 0 0 0 ...
 $ south   : int  0 0 0 0 0 0 0 0 0 0 ...
 $ educ    : int  14 14 14 14 14 14 14 14 13 13 ...
```

```
> head(cbind(wagepan$nr,wagepan$year),50)
      [,1] [,2]
 [1,]   13 1980
 [2,]   13 1981
 [3,]   13 1982
 [4,]   13 1983
 [5,]   13 1984
 [6,]   13 1985
 [7,]   13 1986
 [8,]   13 1987
 [9,]   17 1980
[10,]   17 1981
[11,]   17 1982
[12,]   17 1983
[13,]   17 1984
[14,]   17 1985
[15,]   17 1986
[16,]   17 1987
[17,]   18 1980
[18,]   18 1981
[19,]   18 1982
[20,]   18 1983
[21,]   18 1984
[22,]   18 1985
[23,]   18 1986
[24,]   18 1987
```

Convert the panel data into a structure suitable for the plm() function.

```
> wagepan.panel<-plm.data(wagepan, c("nr","year"))
> summary(wagepan.panel)
       nr             year          agric             black              bus          
 13     :   8   1980   : 545   Min.   :0.00000   Min.   :0.0000   Min.   :0.00000  
 17     :   8   1981   : 545   1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.:0.00000  
 18     :   8   1982   : 545   Median :0.00000   Median :0.0000   Median :0.00000  
 45     :   8   1983   : 545   Mean   :0.03211   Mean   :0.1156   Mean   :0.07592  
 110    :   8   1984   : 545   3rd Qu.:0.00000   3rd Qu.:0.0000   3rd Qu.:0.00000  
 120    :   8   1985   : 545   Max.   :1.00000   Max.   :1.0000   Max.   :1.00000  
 (Other):4312   (Other):1090                                                       
    construc            ent              exper             fin              hisp        
 Min.   :0.000   Min.   :0.00000   Min.   : 0.000   Min.   :0.00000   Min.   :0.000  
 1st Qu.:0.000   1st Qu.:0.00000   1st Qu.: 4.000   1st Qu.:0.00000   1st Qu.:0.000  
 Median :0.000   Median :0.00000   Median : 6.000   Median :0.00000   Median :0.000  
 Mean   :0.075   Mean   :0.01514   Mean   : 6.515   Mean   :0.03693   Mean   :0.156  
 3rd Qu.:0.000   3rd Qu.:0.00000   3rd Qu.: 9.000   3rd Qu.:0.00000   3rd Qu.:0.000  
 Max.   :1.000   Max.   :1.00000   Max.   :18.000   Max.   :1.00000   Max.   :1.000  
```

Random-Effect Estimation

```
# Setup the data
wagepan.panel<-plm.data(wagepan, c("nr","year"))
summary(wagepan.panel)
str(wagepan.panel)

wagepan.re <- plm(lwage ~
educ+black+hisp+exper+exper^2+married+union,data=wagepan.panel,
model="random")
summary(wagepan.re)
```

```
> summary(wagepan.re)
Oneway (individual) effect Random Effect Model
   (Swamy-Arora's transformation)

Call:
plm(formula = lwage ~ educ + black + hisp + exper + exper^2 +
    married + union, data = wagepan.panel, model = "random")

Balanced Panel: n=545, T=8, N=4360

Effects:
                 var std.dev share
idiosyncratic 0.1251  0.3537 0.543
individual    0.1055  0.3248 0.457
theta:  0.6407

Residuals :
   Min. 1st Qu.  Median 3rd Qu.    Max.
-4.5500 -0.1460  0.0253  0.1920  1.5500

Coefficients :
             Estimate Std. Error t-value  Pr(>|t|)
(Intercept) -0.0477025  0.1104704 -0.4318  0.665899
educ         0.1081869  0.0088615 12.2087 < 2.2e-16 ***
black       -0.1409950  0.0476417 -2.9595  0.003098 **
hisp         0.0160861  0.0426212  0.3774  0.705880
exper        0.0579448  0.0025026 23.1537 < 2.2e-16 ***
married      0.0757793  0.0167533  4.5232 6.252e-06 ***
union        0.1100202  0.0179187  6.1400 8.991e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Total Sum of Squares:    657.83
Residual Sum of Squares: 546.44
R-Squared:      0.16934
Adj. R-Squared: 0.16907
F-statistic: 147.903 on 6 and 4353 DF, p-value: < 2.22e-16
```

Fixed-Effect vs. Random-Effect Models

Fixed Effect vs Random Effect Models

- Because fixed effects allows arbitrary correlation between ai and the xitj, while random effects does not, FE is widely thought to be a more convincing tool for estimating ceteris paribus effects.
- Still, random effects is applied in certain situations. Most obviously, if the key explanatory variable is constant over time, we cannot use FE to estimate its effect on y.
- In fact, we had to rely on the RE (or pooled OLS) estimate of the $ in the example given previously.
- However, as emphasized before, we use random effects because we are willing to assume the unobserved effect is uncorrelated with all explanatory variables; this assumption may not always be sensible.

- RE is preferred to pooled OLS because RE is generally more efficient.
- Whenever considering using RE model, one has to give substantial reasons why the assumption $Cov(x_{itj}, a_i) = 0$ is reasonable.
- Hausman (1978) first proposed a test to test the full set of random effects assumptions. The idea is that one uses the random effects estimates unless the Hausman test rejects $Cov(x_{itj}, a_i) = 0$. We will cover Hausman in the R demo.

Berkeley SCHOOL OF INFORMATION