Pooled cross-section and Panel data

Tyler Ransom

Univ of Oklahoma

Apr 9, 2019

Today's plan

1. Intro to pooled cross-section and panel data
    1.1 Pooled cross sections
    1.2 Panel data
    1.3 First differences
    1.4 Fixed effects
2. In-class activity: Work on project

Pooled cross sections

What is a pooled cross section?

- Data obtained by pooling cross sections are very useful
- Can examine trends and conduct policy analysis
- A **pooled cross-section (PCS)** is a repeated cross-section
- e.g. survey is repeated over time with new random samples each year
- e.g. General Social Survey (GSS) and the Current Population Survey (CPS)

How pooled cross sections can help us

- Analyzing pooled cross sections is similar to a single cross section

- Key assumption: each period is a random sample

- Can show us how the mean value of a variable has changed over time …

  … in ways that cannot be explained by observable variables

- e.g. How has fertility changed in ways not explained by educ, LFP?

- PCSs are at the foundation of **difference-in-differences** estimation

- (We'll talk about this in detail next class period)

Example: Fertility over time

- Can use the `fertil1` data set from the `wooldridge` package

- First, count number of observations in each year (and make `year` a factor)

```
> df <- as_tibble(fertil1)

> table(df$year)

 72  74  76  78  80  82  84
156 173 152 143 142 186 177

> df %<>% mutate(year = as.factor(year))
```

Example: Summary statistics

- First let's compute average fertility across years

```
> df %>% group_by(year) %>%
  summarize(avg.fertil=mean(kids))

  year  avg.fertil
  <fct>      <dbl>
1 72          3.03
2 74          3.21
3 76          2.80
4 78          2.80
5 80          2.82
6 82          2.40
7 84          2.24
```

Example: Regression model

```
> est <- lm(kids ~ educ + age + I(age^2) + year, data=df)

> stargazer(est, type="text")
===============================================
                          kids
-----------------------------------------------
educ                    -0.119***
                         (0.018)
age                     0.501***
                         (0.141)
I(age2)                 -0.005***
                         (0.002)
year74                    0.243
                         (0.175)
year76                   -0.146
                         (0.181)
year78                   -0.096
                         (0.184)
```

Example: Regression model

```
year80                            -0.077
                                  (0.184)
year82                           -0.428**
                                  (0.174)
year84                           -0.553***
                                  (0.177)
Constant                         -6.862**
                                  (3.096)
----------------------------------------------
Observations                       1,129
R2                                 0.091
Adjusted R2                        0.084
Residual Std. Error         1.583 (df = 1119)
F Statistic             12.463*** (df = 9; 1119)
==============================================
Note:                  *p<0.1; **p<0.05; ***p<0.01
```

Example: discussion

- The above analysis tells us fertility is going down
- Even when we hold education fixed
- (So, we can rule out that fertility $\downarrow$ because education $\uparrow$)
- (Also can rule out that fertility $\downarrow$ because age $\uparrow$)
- We would instead call this a “secular” decline in fertility
- This finding could be of interest to demographers and others

Panel data

What is panel data?

- With a panel data set, the same units are sampled in 2+ time periods
- For each unit i (individual, school, etc.) we have 2+ years of data
- y will be correlated over time within unit i; we must correct for this
- In PCS data, this doesn't happen (each year is a new random sample)
- **Main benefit of panel data:** Can control for persistent unobservables
- This is very useful for policy analysis

Some particulars

- **Balanced panel:** we observe the same time periods for each unit
- For simplicity, we'll assume balanced panels, but this rarely holds in reality
- People leave the survey, firms close down or merge, etc.
- Attrition can severely bias estimates—always keep this in mind
- Notation: i is a unit, t is time, y_{it} is outcome, x_{itk} is a covariate

Two-period panels

- For simplicity, start with a two-period panel
- Along with the observed data $(x_{it1}, \ldots, x_{itk}, y_{it})$ we draw unobserved factors
- Put these factors into two categories:
    1. a_i a component that changes over i but not t (e.g. innate ability)
    2. u_{it} unobservables that change across time (a.k.a. idiosyncratic errors)

Storing panel data

- There are two ways to store panel data:

  1. **wide**, i.e. with each year as a separate variable

  2. **long**, i.e. with each year as a separate row

- Long format is by far the most common

- When using long format, make sure you order chronologically within unit

Long format

```
> df <- as_tibble(gpa3) %>%
select(id,term,trmgpa,season)

> head(df)

      id  term trmgpa season
1    22.     1   1.50      0
2    22.     2   2.25      1
3    35.     1   2.20      0
4    35.     2   1.60      1
5    36.     1   1.60      0
6    36.     2   1.29      1
```

Wide format

```
> df <- as_tibble(vote2) %>%
select(state, district, vote88,
vote90, inexp88, inexp90)

> head(df)

  st dis vo88 vo90 inexp88 inexp90
1 AL   2   94   51 234923. 596096.
2 AL   3   65   74 679297. 176550.
3 AL   7   68   71 328296. 238446.
4 AK   1   62   52 626377. 564759.
5 AZ   2   73   66  99607. 112373.
6 AZ   3   69   57 319690. 225149.
```

Converting from wide to long

- To convert from wide to long, use `gather()` and `spread()`

```
> df.wide <- as_tibble(vote2) %>%
select(state, district, vote88, vote90, inexp88, inexp90)

> df.long <- df.wide %>%
     gather(variable,value,-state,-district) %>%
     mutate(year = parse_number(variable)) %>%
     mutate(variable =  gsub("\\d","",x = variable)) %>%
     spread(variable,value)

> head(df.long)
  state district  year   inexp  vote
1 AK           1   88. 626377.   62.
2 AK           1   90. 564759.   52.
3 AL           2   88. 234923.   94.
4 AL           2   90. 596096.   51.
5 AL           3   88. 679297.   65.
6 AL           3   90. 176550.   74.
```

Converting from wide to long

- In the previous slide, you can use the same code for other applications
- Key idea: put minus in front of time-invariant variables in `gather()`
- everything else should work as expected
- **Always double-check that your long data matches up with your wide data**
- e.g. make sure that vote in 1988 has same value in both data sets

First differences

Estimating a two-period model

- Suppose we have a balanced two-period panel. A general equation is

$$y_{it} = \beta_0 + \delta_0 d2_t + \beta_1 x_{it} + a_i + u_{it}, \quad t = 1, 2.$$

- $d2_t$ is a dummy indicating the second period
- a_i is the unobserved unit effect (a.k.a. unobserved heterogeneity)
- u_{it} is the unobserved idiosyncratic error
- We are interested in estimating β_1, the partial effect of x on y. How can we?

Option 1: Pooled OLS

- We could estimate a similar model by pooled OLS:

$$y_{it} = \beta_0 + \delta_0 d2_t + \beta_1 x_{it} + \nu_{it}, \quad t = 1, 2.$$

  where $v_{it} = a_i + u_{it}$

- We would simply regress y on $d2$ and x

- Two issues arise:

  1. ν_{it} not i.i.d. (but can correct this with cluster robust SEs)

  2. x_{it} might be correlated with ν_{it} through a_i

  - (2) is known as **heterogeneity bias**. Can fix this by taking **first differences**

Option 2: First Differences

- Let's rewrite the original model, stacking time periods:

$$y_{i2} = \beta_0 + \delta_0 \underbrace{d2_2}_{=1} + \beta_1 x_{i2} + a_i + u_{i2}$$

$$y_{i1} = \beta_0 + \delta_0 \underbrace{d2_1}_{=0} + \beta_1 x_{i1} + a_i + u_{i1}$$

- Now subtract:

$$\Delta y_i = \delta_0 + \beta_1 \Delta x_i + \Delta u_i$$

- We are left with a cross-sectional-looking model, and a_i is now gone

First Differences

- We simply estimate the differenced equation by (pooled) OLS
- This is known as the **first-difference estimator**
- Note: Need x_{it} to vary with time!
- Note: δ_0 is difference in intercept; this can be an interesting parameter
- β_1 is same as before; measures causal effect of x on y
- Interpretation also same; we differenced just to get rid of a_i

Fixed effects

Fixed effects

- Differencing is one method of eliminating a_i
- Alternatively, can use the **fixed effects** or **within** transformation
- We subtract from y and x their within-i time averages
- In the simple model with only x_{it}:

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

- Average this equation across t to get

$$\bar{y}_i = \beta_0 + \beta_1 \bar{x}_i + a_i + \bar{u}_i$$

where $\bar{y}_i = T^{-1} \sum_{t=1}^{T} y_{it}$ is a "time average" for unit i, and so forth

Fixed effects

- Subtract the time-averaged equation from other time periods:

$$y_{it} - \overline{y}_i = \beta_1(x_{it} - \overline{x}_i) + (u_{it} - \overline{u}_i)$$

- As with the FD equation, this equation is free of a_i
- As with FD, simply estimate the differenced model with pooled OLS
- As with FD, interpretation is on the untransformed model
- Trivia: the time-averaged eq. is sometimes called the "between equation"

First differences or Fixed effects?

- Comparison of FD and FE is nuanced
- FD is better if there is high serial correlation in u_{it}
- FD is better if $T > N$
- FE is much more commonly used in applied micro research
- If $T = 2$ the FD and FE estimates are identical

Dummy Variable Regression

- It turns out that FE is the same as **dummy variable regression**
- i.e., put a dummy for each unit
- Allows us to estimate $\hat{a}_i$ for each unit
- But not practical for panels with large cross sections
- R^2 is usually very, very high and not meaningful

Fixed effects in R

- In R, use `plm( )`, which stands for Panel Linear Model

- Requires data to be in “long” form and unit index to be one variable

```
# combine unit identifiers into one variable
df.long %<>% unite(stdist, state:district, sep = "") %>%
             mutate(stdist=as.factor(stdist))
est.pols <- plm(vote ~ log(inexp), data = df.long,
index = c("stdist","year"), model = "pooling")
est.fd <- plm(vote ~ log(inexp), data = df.long,
index = c("stdist","year"), model = "fd")
est.fe <- plm(vote ~ log(inexp), data = df.long,
index = c("stdist","year"), model = "within")
# dummy variable regression: note use of lm() and not plm()
est.dv <- lm(vote ~ log(inexp) + stdist, data = df.long)
```

Comparison of results

```
> stargazer(est.pols,est.fd,est.fe,est.dv,keep.stat=c("rsq"),type="text",
            column.labels=c("pols","fd","fe","dv"))

====================================================
                          vote
                    panel linear              OLS
               pols      fd         fe         dv
               (1)       (2)        (3)        (4)
----------------------------------------------------
log(inexp) -5.056***  -6.052***  -6.052***  -6.052***
            (0.666)    (1.415)    (1.415)    (1.415)
Constant   129.029***                       137.470***
            (8.493)                          (19.466)
----------------------------------------------------
R2           0.135      0.071      0.090      0.733
====================================================
Note:                        *p<0.1; **p<0.05; ***p<0.01
```

Standard Errors

- With panel data, we have to worry about serially correlated errors
- Simple fix: use cluster-robust SEs
- Clustering should be done at the unit level (i.e. person, state, firm, etc.)

```
> coef_test(est.fe, vcov = "CR1", cluster = "individual")
        Coef Estimate   SE d.f. p-val (Satt) Sig.
1 log(inexp)    -6.05 1.53   38       <0.001  ***
```

- Clustering makes the SE a bit larger
- Clutering the pooled OLS estimate also increases its SE (to 1.15 from 0.67)

Standard Errors

- You can provide clustered SEs to `stargazer` as follows:

```
> clust.po <- coef_test(est.pols, vcov = "CR1",
                        cluster = "individual")

> clust.fe <- coef_test(est.fe, vcov = "CR1",
                        cluster = "individual")

> stargazer(est.pols,est.fe,
            se=list(clust.po$SE,clust.fe$SE),
            type="text")
```