Pooled cross-section and Panel data

Tyler Ransom

Univ of Oklahoma

Apr 11, 2019

Today's plan

1. Brief Review
2. Random effects
3. In-class activity: estimating panel models

Brief Review

Two kinds of “panel” data

1. Pooled cross sections

2. True panel data

Usefulness of pooled cross sections

- Pooled cross sections can tell us about how key variables are trending

- Can also tell us about causality (topic of next lecture)

Usefulness of panel data

- Panel data allows us to explicitly control for persistent unit-specific u's

- This makes it easier to obtain causal effects

Panel data estimators

- There are four potential estimators that can be used on panel data:

  1. Pooled OLS (ignores persistent u's! Don't use)
  2. First differences
  3. Fixed effects

- FD and FE are equivalent when $T = 2$
- The fourth, **random effects**, is our topic for today

Standard errors

- It’s most appropriate to report “clustered” standard errors

- These correct for within-unit serial correlation

- Also are robust to heteroskedasticity

Random Effects

Random Effects

- Suppose we start with the same equation as before:

$$y_{it} = \delta_t + \beta_1 x_{it1} + \cdots + \beta_k x_{itk} + a_i + u_{it}, \qquad t = 1, 2, ..., T$$

- δ_t represents different time intercepts (drop for simplicity)

- With FE, we would difference out the “within” data to remove a_i

- With **random effects (RE)**, we leave a_i in the error term

- Then, account for serial correlation in $\nu_{it} = a_i + u_{it}$

- Do so using a **Generalized Least Squares (GLS)** procedure

Random Effects

- What is the implication of leaving a_i in the error term?
- Now, we can include time-invariant x_i's in our model
- With FE, we could tell x_i's apart from a_i
- Now, we can separate the two
- But this by assumption, and it doesn't come cheap

Random Effects and Policy Evaluation

- Typically, it's not useful to use RE for policy analysis
- The reason is, we want a_i to be correlated with the policy
- e.g. factors with less able employees apply for hiring grants
- Sometimes, RE can be convincing if we have good x_i's
- Because as we add more x_i's, more is taken out of a_i

Random Effects Assumptions

- Most controversial assumption for RE is $Cov(x_{itj}, a_i) = 0$
- When this assumption holds, RE is consistent **and** efficient
- When it doesn't hold, it is inconsistent
- Best way to see if RE assumption makes sense:
- Estimate pooled OLS, RE, and FE and see how different they are

Random Effects Estimation

- To estimate RE, we use a Feasible GLS estimator

- Quasi-demean the data and estimate with pooled OLS:

$$\begin{aligned} y_{it} - \theta\bar{y}_i = \beta_0 (1 - \theta) + \beta_1 (x_{it1} - \theta\bar{x}_{i1}) + \cdots \\ + \beta_k (x_{itk} - \theta\bar{x}_{ik}) + (\nu_{it} - \bar{\nu}_i) \end{aligned}$$

  where

$$\theta = 1 - \sqrt{\frac{\sigma_u^2}{\sigma_u^2 + T\sigma_a^2}}$$

- The idea is similar to removing serial correlation with quasi-demeaning

RE vs. FE

- When $\theta = 0$, we get the pooled OLS estimates

- When $\theta = 1$, we get the fixed effects estimates

- Usually $0 < \theta < 1$

- If $\theta = 0$ that means there is no unobserved heterogeneity

- This is highly unlikely in typical applications

- For this reason, people usually favor FE to get causal effects

RE estimation in R

- RE estimation is another option in the `plm()` function

- Repeating the same analysis of vote shares as last time:

```
> 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)

> df.long %<>% unite(stdist, state:district, sep = "") %>%
               mutate(stdist=as.factor(stdist))
```

RE estimation in R

```
> est.pols <- plm(vote ~ log(inexp), data = df.long,
index = c("stdist","year"), model = "pooling")

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

> est.re <- plm(vote ~ log(inexp), data = df.long,
index = c("stdist","year"), model = "random")

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

> est.fe <- plm(vote ~ log(inexp), data = df.long,
index = c("stdist","year"), model = "within")

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

RE estimation in R

```
# To see what theta is estimated to be:
> est.re$ercomp$theta
       id
0.3334773

> stargazer(est.pols,est.re,est.fe,
            se=list(clust.po$SE,clust.re$SE,clust.fe$SE),keep.stat=c("rsq"
            type="text",column.labels=c("pols","re","fe"))

==========================================
                          vote
                pols       re        fe
------------------------------------------
log(inexp) -5.056***  -5.201***  -6.052***
             (1.147)    (1.044)    (1.531)
------------------------------------------
R2            0.135      0.122      0.090
==========================================
```