Notes for POLS 607: Panel data*

Casey Crisman-Cox

Spring 2025

*These notes are for personal use only and not for distribution. They are likely rife with errors, typos, and are not referenced. This is not my own work, but simply notes for me to lecture from.

Contents

1 Review: Linear panel model basics

We will start with a review of the standard panel data models. Before getting into that we'll need some assumptions. The first of which describes what a basic panel is:

Assumption A1 *The data generating process is linear-in-the-parameters, such that*

$$y_{it} = \alpha_i + \beta' x_{it} + \gamma' z_i + \varepsilon_{it}.$$

Usually we think of i as "units" (individuals, states, countries, dyads, etc) and t as "within-unit" observations (typically time, but could be multiple individuals within a unit, etc). We will let $i = 1, \ldots, N$, $t = 1, \ldots, T$ and NT be the total number of observations. To make exposition easier, we will often assume a "balanced" panel where T is the same for each i. When necessary, we will talk about cases where this distinction matters.

Neither x_{it} nor z_i contain a constant term, instead α_i reflects a general situation where each unit has its own constant term. More on this to come, but for now we will simplify that further and assume that $\alpha_i = \alpha$ for all i (**Assumption A1.A**). Additionally, x_{it} is a variable that changes both across and within units, while z_i is constant within units but variable across units.

Given this setup, we are unlikely to have independent observations. After all, if our panel is a collection of N separate time series, then assuming independence is a pretty long stretch from the start, but we will typically want a type of independence assumption

Assumption A2 *Each unit* $(x_i, z_i, \varepsilon_i)$ *is drawn iid.*

The notation $x_i = (x_{i1}, \ldots, x_{iT})$ used here refer to all observations of x_{it} within unit i. Here we are making the assumption that each block of observations is independent of the rest and that the units are drawn from some population process. This is not perhaps super convincing in some cases, but it is a convenience assumption and a start.

We are not currently making any assumptions about the dependency structure on the within-unit observations. However, we will need to make some kind of exogeneity assumption (as always)

Assumption A3 *There is strict exogeneity within units,* $\mathrm{E}[\varepsilon_{it}|x_i, z_i] = 0$.

This tells us that within-each unit, we assume that the error term is independent of the observables (i.e., no unobserved confounding within units).

Let $\mathbf{X} = [1\ X\ Z]$ and let $\theta = (\alpha, \beta, \gamma)'$, then the *pooled* OLS estimator for the panel model is

$$\begin{aligned}\hat{\theta}_p &= \left(\frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T}\mathbf{x}_{it}\mathbf{x}_{it}'\right)^{-1}\frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T}\mathbf{x}_{it}y_{it} \\ &= (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'y,\end{aligned}$$

or (by A1.A)

$$\hat{\theta}_p = \theta + \left(\frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T}\mathbf{x}_{it}\mathbf{x}_{it}'\right)^{-1}\frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T}\mathbf{x}_{it}\varepsilon_{it}.$$

By strict exogeneity within unit (A3) and independence across units (A2) we have

$$\mathrm{E}[\varepsilon_{it}|x_i, z_i] = \mathrm{E}[\varepsilon_{it}|\mathbf{X}] = 0.$$

Thus we can apply iterated expectations to get

$$\begin{aligned}\mathrm{E}[\mathrm{E}[\hat{\theta}_p|\mathbf{X}]] &= \theta + \mathrm{E}\left[\left(\frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T}\mathbf{x}_{it}\mathbf{x}_{it}'\right)^{-1}\frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T}\mathbf{x}_{it}\,\mathrm{E}[\varepsilon_{it}|\mathbf{X}]\right] \\ &= \theta + 0.\end{aligned}$$

Which gives us our first property,

Property A1 *Under Assumptions A1.A, A2, and A3 the pooled estimator $\hat{\theta}_p$ is unbiased, if it exists.*

Note there will be times when we get to dynamics where strict within-unit exogeneity doesn't make sense. For now, we'll go with it.

Further, we will impose a rank condition

Assumption A4 *The matrix $\mathbf{Q} = \mathrm{E}\left[\frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_{it}\mathbf{x}_{it}'\right] < \infty$ has full rank.*

With this assumption, we assert that the DGP for the T within-unit observations are well-behaved and $X_i'X_i$ has full rank for each unit. Likewise, since it is in a quadratic form, the matrix will be positive definite under this assumption.

Finally, we will include some hand-wavy moment conditions on the data and errors.

Assumption A5 *Additional moment assumptions that allow us to apply a central limit theorem (CLT).*

For review, let's briefly restate these theorems along with some other useful results

Theorem 1 (Weak Law of Large Numbers) *Let $X_1, \ldots, X_N$ be an iid sequence of random variables, where each $X_i \in \mathbb{R}^k$ has a finite absolute first moment $\mathrm{E}[X_i] < \infty$. Then $\frac{1}{N}\sum_{i=1}^{N} X_i \xrightarrow{p} \mathrm{E}[X_i]$.*

Theorem 2 (Central Limit Theorem) *Let $X_1, \ldots, X_N$ be iid random variables with expected value μ and variance Ω (both finite), then for all $x \in \mathbb{R}$*

$$\sqrt{N}\left(\frac{1}{N}\sum_{i=1}^{N} X_i - \mu\right) \xrightarrow{d} N(0, \Omega).$$

Theorem 3 (Continuous mapping theorem) *Consider a sequence of random variables $X_n = X_1, \ldots X_N$ and a continuous function g*

- *If $X_n \xrightarrow{d} X$, then $g(X_n) \xrightarrow{d} g(X)$*
- *If $X_n \xrightarrow{p} X$, then $g(X_n) \xrightarrow{p} g(X)$*

Theorem 4 (Functions preserve iid) *Let g be a continuous function and let X and Y be random variables*

- *If X and Y are independent, then $g(X)$ and $g(Y)$ are independent random variables*
- *If X and Y are identically distributed, then $g(X)$ and $g(Y)$ are identically distributed*

Theorem 4 is a deceptively powerful result. We can now say that since we know x_i and ε_i are each iid, then $g(x_i)$ and $g(\varepsilon_i)$ will retain these properties for continuous g.

Theorem 5 (Slutsky's Theorem) *Let X_i and Y_i be sequences of random variables.*

1. *If $X_n \xrightarrow{d} X$ and $Y_n \xrightarrow{p} y$ (where y is a constant), then*
   - $X_n Y_n \xrightarrow{d} Xy$
   - $Y_n^{-1} X_n \xrightarrow{d} (y^{-1})X$, *if y^{-1} exists*
2. *If $X_n \xrightarrow{p} x$ and $Y_n \xrightarrow{p} y$ (where x and y are constants), then*
   - $X_n Y_n \xrightarrow{p} xy$
   - $Y_n^{-1} X_n \xrightarrow{p} (y^{-1})X$, *if y^{-1} exists*

We can now apply a standard LLN type argument

1. Let's start with the expression $\frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_{it}\mathbf{x}_{it}'\right)$, what happens here as N grows? Assumption A4 will let us know that $\mathrm{E}\left[\frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_{it}\mathbf{x}_{it}'\right]$ is finite. Likewise, Assumption A2 tells us that x_i and x_j are iid for $i \neq j$. Note that $\frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_{it}\mathbf{x}_{it}'$ and $\frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_{jt}\mathbf{x}_{jt}'$ are functions of iid random variables and so are themselves iid by Theorem 4. As such we can apply the LLN to get

$$\frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_{it}\mathbf{x}_{it}'\right) \xrightarrow{p} \mathrm{E}\left[\frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_{it}\mathbf{x}_{it}'\right] = \mathbf{Q}$$

2. By Assumption A4, $\mathbf{Q}$ has full rank and will be positive definite. Because it's positive definite the inverse function will be continuous and we can apply the CMT, to get

$$\left[\frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_{it}\mathbf{x}_{it}'\right)\right]^{-1} \xrightarrow{p} \mathbf{Q}^{-1}.$$

3. Now consider $\frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_{it}\varepsilon_{it}\right)$. Using Assumption A2 again we know that $(x_i'\varepsilon_i)_{i=1}^{N}$ represents N iid random vectors. We can apply the LLN to see that

$$\frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_{it}\varepsilon_{it}\right) \xrightarrow{p} \mathrm{E}\left[\frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_{it}\varepsilon_{it}\right],$$

where

$$\begin{aligned}\mathrm{E}\left[\frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_{it}\varepsilon_{it}\right] &= \frac{1}{T}\sum_{t=1}^{T}\mathrm{E}\left[\mathbf{x}_{it}\varepsilon_{it}\right] \\ &= \mathrm{E}_x\left[\mathrm{E}\left[\mathbf{x}_{it}\varepsilon_{it}|\mathbf{X}\right]\right] \\ &= 0\end{aligned}$$

by Assumption A2–A3.

4. We can now apply Slutsky's theorem to say:

$$\left(\frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T}\mathbf{x}_{it}x_{it}'\right)^{-1}\frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T}\mathbf{x}_{it}\varepsilon_{it} \xrightarrow{p} \mathbf{Q}^{-1}0 = 0.$$

5. Finally, since the sum operator is continuous we can again apply the continuous mapping theorem to get

$$\begin{aligned}\hat{\theta}_p &= \theta + \left(\frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T}\mathbf{x}_{it}x_{it}'\right)^{-1}\frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T}\mathbf{x}_{it}\varepsilon_{it} \\ &\xrightarrow{p} \theta + 0 = \theta\end{aligned}$$

Property A2 *Under A1.A and A2–A4, as $N \to \infty$ the pooled estimator exists. The pooled estimator is consistent for θ.*

Asymptotic normality follows in a similar way, but let's recap it too,

1. Rewrite the estimator to look like Theorem 2.

$$\sqrt{N}(\hat{\theta}_p - \theta) = \left(\frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} \mathbf{x}_{it} \mathbf{x}_{it}' \right)^{-1} \frac{\sqrt{N}}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} \mathbf{x}_{it} \varepsilon_{it}$$

2. From above we know

$$\frac{1}{T} \sum_{t=1}^{T} \mathbf{x}_{it} \mathbf{x}_{it}' \xrightarrow{p} \mathrm{E}\left[\frac{1}{T} \sum_{t=1}^{T} \mathbf{x}_{it} \mathbf{x}_{it}' \right] = \mathbf{Q}$$

3. The remaining term

$$\sqrt{N} \left(\frac{1}{N} \sum_{i=1}^{N} \left(\frac{1}{T} \sum_{t=1}^{T} \mathbf{x}_{it} \varepsilon_{it} \right) \right)$$

   looks like the CLT, right? and we know that

$$\mathrm{E}\left[\frac{1}{T} \sum_{t=1}^{T} \mathbf{x}_{it} \varepsilon_{it} \right] = 0,$$

   from above. So, we can apply that to get to

$$\sqrt{N} \left(\frac{1}{N} \sum_{i=1}^{N} \left(\frac{1}{T} \sum_{t=1}^{T} \mathbf{x}_{it} \varepsilon_{it} \right) \right) \xrightarrow{d} N\left(0, \Sigma_N\right).$$

   We will assume that $\Sigma_N = \mathrm{Var}\left(\frac{1}{T} \sum_{t=1}^{T} \mathbf{x}_{it} \varepsilon_{it} \right)$ exists under Assumption A5.

4. Finally, we can combine terms using Slutsky's theorem to get

$$\sqrt{N}(\hat{\theta}_p - \theta) \xrightarrow{d} N(0, \mathbf{Q}^{-1} \Sigma_N \mathbf{Q}^{-1}).$$

All together this gives us our next property:

Property A3 *Under A1.A and A2–A5 the pooled estimator $\hat{\theta}_p$ is asymptotically normal such that*

$$\sqrt{N}(\hat{\theta}_p - \theta) \xrightarrow{d} N(0, \mathbf{Q}^{-1} \Sigma_N \mathbf{Q}^{-1})$$

This is a format you should be used to seeing by now. If we assumed within-unit independence and homoskedasticity we would get the classic OLS variance, how likely do you think that is?

We can estimate the asymptotic variance of θ using sample counterparts:

$$\begin{aligned}
\text{avar}\left(\hat{\theta}_p\right) &= \frac{1}{N}\mathbf{Q}^{-1}\Sigma_N\mathbf{Q}^{-1} \\
\widehat{\text{avar}}\left(\hat{\theta}_p\right) &= \frac{1}{N}\hat{\mathbf{Q}}^{-1}\widehat{\Sigma}_N\hat{\mathbf{Q}}^{-1} \\
\hat{\mathbf{Q}} &= \frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T}\mathbf{x}_{it}\mathbf{x}_{it}' \\
\widehat{\Sigma}_N &= \frac{1}{N}\sum_{i=1}^{N}\left[\frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_{it}\hat{\varepsilon}_{it}\right]\left[\frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_{it}\hat{\varepsilon}_{it}\right]' \\
\widehat{\text{avar}}\left(\hat{\theta}_p\right) &= \left[\sum_{i=1}^{N}\mathbf{x}_i'\mathbf{x}_i\right]^{-1}\left(\sum_{i=1}^{N}\mathbf{x}_i'\varepsilon_i\varepsilon_i'\mathbf{x}_i\right)\left[\sum_{i=1}^{N}\mathbf{x}_i'\mathbf{x}_i\right]^{-1}.
\end{aligned}$$

This variance matrix is called the *cluster-robust* or *clustered* variance matrix. The square root of the diagonal provides *clustered standard errors*. Note that the clustered variance matrix allows for *arbitrary* correlation among the errors within each unit. We haven't imposed any structure on them, not even stationarity. This is a powerful result that makes the clustered matrix very popular.

A warning, this sandwich is valid for large-N asymptotics. That's all we've done so far. An intutitive way to think about this is to note, that Σ_N is estimated by computing the variance within each unit and averaging over units As such, this estimator is only asymptotically valid **in** N. The standard rule of thumb is you have less than 50 units, the clustered matrix is probably not reliable and you may be better off with basic robust standard errors (if NT is large enough), or other alternatives based on large-T asymptotics that we'll get to later.

Some recent work has considered the issue with a small number of clusters. Some proposals here include:

1. A fixed number of clusters correction: $\frac{N}{N-1}\widehat{\text{avar}}\left(\hat{\theta}_p\right)$.
2. Wild-bootstrap which one of your classmates will present on later

Another approach is to do a clustered or block bootstrap. This *should* give you similiar results to the asymptotic standard errors, but they can diverge for any number of reasons.

To review, a bootstrap is a tool that relies on the *empirical distribution* to estimate the true distribution. Consider a sample $y_1, \ldots, y_N \overset{iid}{\sim} F(y)$ where F is an unknown CDF with some parameter of interest θ what is a function $\theta = T(F(y))$. We can consider the empirical

distribution function (EDF) F_N as an approximation of F.

$$F_N(y) = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}(y_i \leq y),$$

which is a discrete distribution that puts probability $1/N$ on each observation. We want to be able to use this CDF to say something about true CDF and more importantly for us usually, some function of it T.

For example, if T is the expected value as in

$$T(F(y)) = \mathrm{E}[y] = \int y f(y) dy = \mu_y$$

and substituting the EDF gives us

$$T(F_N(y)) = \sum_{i=1}^{N} y_i f_N(y_i) = \frac{1}{N}\sum_{i=1}^{N} y_i = \hat{\mu}_y,$$

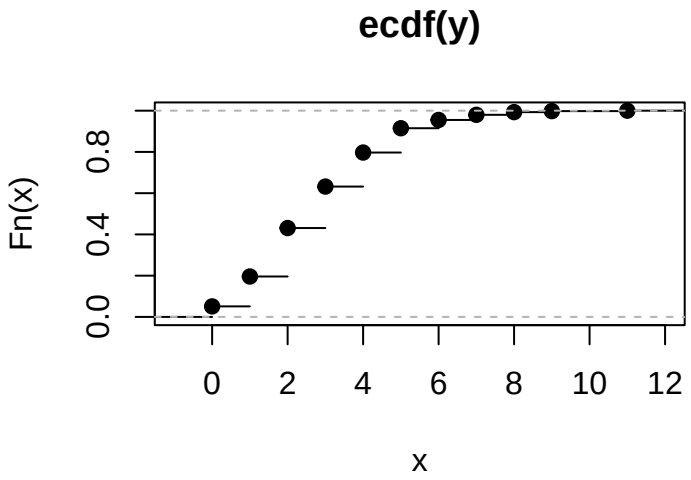
which better known as the sample mean. If we wanted to know the variance of this estimate without knowing anything else about the true distribution, we would ask, well do we think the EDF is a good approximation of the CDF? So long we say yes to that, then we can use the EDF as if it were the CDF and draw "new" samples from it.

You may be familiar with how to sample from CDFs, typically we look for a solution based on inverse uniform sampling. This approach has us

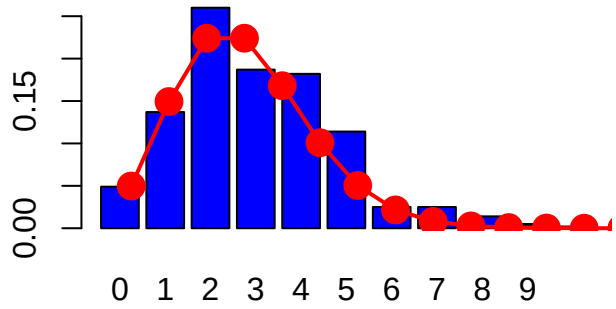
1. Generate U which is a length-N vector of draws from the standard uniform
2. Generate $y^* = F^{-1}(U)$ which is our new sample.

In the case of the EDF, we always have a step function and so inversion is just the quantile function. Which is nice.

```
set.seed(10)
y <- rpois(1000, lambda=3)
plot(ecdf(y))
```

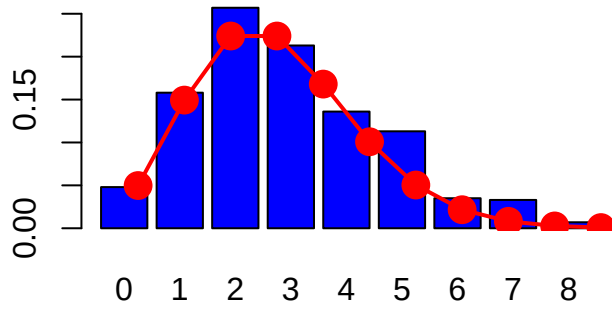



```
U <- runif(1000)
y2 <- quantile(y, probs=U, type=1)
barplot(table(y2)/1000, col="blue")
points(dpois(seq(0, 20,by=1), lambda=3), col="red", pch=16, cex=2)
lines(dpois(seq(0, 20,by=1), lambda=3), col="red", lwd=2)
```



This looks a little convoluted for what it actually is. Drawing samples from the EDF, is just a fancy way to say resample the data with replacement!

```
y3 <- sample(y, size=1000, replace=TRUE)
barplot(table(y3)/1000, col="blue")
points(dpois(seq(0, 20,by=1), lambda=3), col="red", pch=16, cex=2)
lines(dpois(seq(0, 20,by=1), lambda=3), col="red", lwd=2)
```



So that describes the ordinary bootstrap. When dealing with clustered data, however, we don't think that independence necessarily holds at the level of the individual observation. Instead, we sample at the level of the unit, reflecting Assumption A2. For bootstrap iteration $b = 1, 2, \ldots, B$:

1. Resample entire units in the data $(y_i, \mathbf{x}_i)$ with replacement to create a new data set
2. Fit the model using this new dataset
3. Save the estimates $\hat{\theta}^b$

We can then estimate the variance as

$$\frac{1}{B}\sum_{b=1}^{B}\left(\hat{\theta}^b - \hat{\bar{\theta}}\right)\left(\hat{\theta}^b - \hat{\bar{\theta}}\right)',$$

where $\hat{\bar{\theta}} = B^{-1}\sum_{b=1}^{B}\hat{\theta}^b$.

One issue with the standard clustered bootstrap comes into play when we're dealing with unbalanced panels. Now the sample size is changing with each bootstrap iteration, this can lead to numerical oddities. The more unbalanced, the more pronounced the issue. An alternative approach is called a Bayesian bootstrap, which comes from Rubin (1981). Here we think about a different sampling approach where we assign weights to each unit. In the traditional clustered bootstrap these weights are integers 0, 1, 2, ..., N, depending on how many times that unit appears in the data. These integer weights will sum to the number of units N but not the total sample size.

In the Bayesian version, we smooth the weights. The basic insight here is that the integer weights form a multinomial distribution, in Bayesian stats, the Dirichlet distribution is often paired with the multinomial as its continuous counterpart (glossing over details here). Visually we can think about this with the following

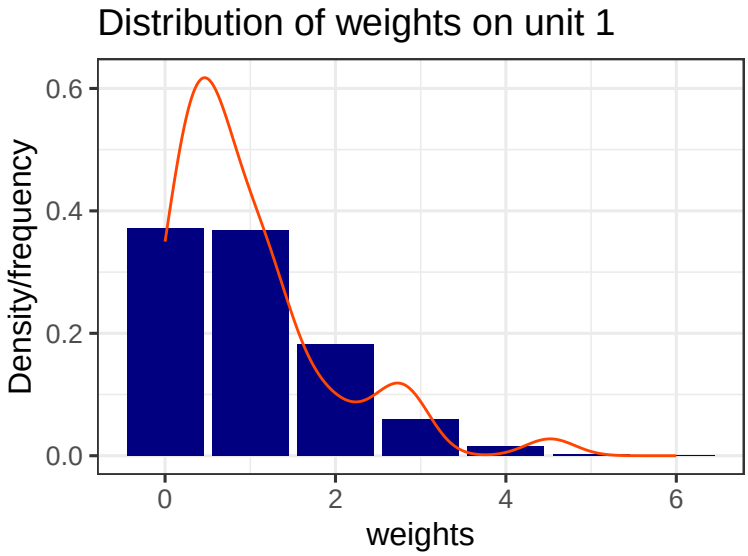
```
set.seed(1)
N <- 50
B <- 10000

## draw a bunch of samples. Each row is a sample
multinomial <- matrix(sample(1:N, size=N*B, replace=TRUE), nrow=B)
m1 <- rowSums(multinomial==1)

gamma11 <-  matrix(rexp(N*B), nrow=B)
dircihlet <- gamma11/rowSums(gamma11)
```

```r
d1 <- dircihlet[1,]*N

ggplot()+
  geom_bar(aes(x=m1, y=after_stat(prop)), fill="navyblue")+
  geom_density(aes(x=d1), color="orangered")+
  ggtitle("Distribution of weights on unit 1")+
  xlab("weights")+
  theme_bw(12)+
  ylab("Density/frequency")
```



Note the trick for generating Dirichlet weights:

1. N Gamma random variables with parameters α_i and β divided by their sum is distributed Dirichlet$(\alpha_1, \ldots, \alpha_N)$
2. Because we want a uniform probability of picking any unit, we'll use a Gamma(1,1), which is an Exponential(1)

So for iteration bootstrap b,

1. Draw N values from an Exponential(1) divide these by their sum to get the weights w.
2. Repeat each w_i T_i times where T_i is the number of observations within unit i to get weights for each observation.

So now we have some useful results for the panel model including is asymptotic variance matrix, along with two different bootstrap procedures that may help us if we don't feel confident using asymptotic results. However, one of the main motivations for panel data

is that it gives us repeated looks at individual units. So far we've only treated that as a nuisance that we correct for in the standard errors, but it actually provides us with some important ways to think about omitted variables and unobservables that are constant within units. This kind of unit-level heterogeneity has not yet appeared in our discussion as we have so far pooled all of our units together in estimation. The pooled model restricts us to only consider obervable differences across units (through z_i). To consider heterogeneity outside of the observables, we will need to expand our thinking.

1.1 Random effects

The random effects (RE) model also starts with Assumption A1

Assumption A1

$$y_{it} = \alpha_i + \beta' x_{it} + \gamma' z_i + \varepsilon_{it}$$

Where we diverge from the pooled model is that we will allow for heterogeneity across groups to enter in through α_i now such that we'll replace this model with

Assumption A1.R

$$y_{it} = \alpha + \beta' x_{it} + \gamma' z_i + \alpha_i + \varepsilon_{it}$$

Here the heterogeneity is modeled as an overall constant with random, unit-specific differences. This unit-level heterogeneity is time-invariant and contains any invariant factors that are not included in z_i. In the RE world, each α_i is an iid draw from some distribution (thus the name), making it a stochastic component like the error term. Strict within-unit exogeneity (Assumption A3.R) in this context means

Assumptions A2.R, A3.R, A4.R, & A5.R

$$\begin{aligned}
\mathrm{E}[\varepsilon_{it}|\mathbf{x}_i] &= 0 \\
\mathrm{E}[\alpha_i|\mathbf{x}_i] &= 0 \\
\mathrm{E}[\alpha_i\varepsilon_i|\mathbf{x}_i] &= 0 \\
\mathrm{E}[\varepsilon_{it}\varepsilon_{jt'}|\mathbf{x}_i] &= 0 \\
\mathrm{E}[\alpha_i^2|\mathbf{x}_i] &= \sigma_\alpha^2 \\
\mathrm{E}[\varepsilon_{it}^2|\mathbf{x}_i] &= \sigma_\varepsilon^2
\end{aligned}$$

for all $i \neq j$ or $t \neq t'$.

Note that we also slid a few homoskedasticity assumptions (A4.R) and an iid assumption within units (A2.R). This means that we are in a world that is in some ways more restrictive

than the pooled model. So we've generalized a bit through the unit-specific heterogeneity, but at the cost of some rather strong additional modeling assumptions. To see the restrictiveness of this model over the pooled model, consider that all the within-unit correlation here comes from the presence of α_i. As such it takes a very specific forms. In contrast, the basic model allowed for arbitrary within-unit correlation.

Additionally, note that in the pooled model we assumed that $\alpha_i = 0$, but what if instead we just assumed it was a set of time-invariant omitted variables? If the random effects assumptions are true then $\mathrm{E}[\alpha_i|\mathbf{x}_i] = 0$ and so the omitted variables are uncorrelated with the observables. This means that pooled estimator $\hat{\theta}_p$ is unbiased, consistent, and asymptotically normal for the the random effects model. However, it won't be the most efficient estimator for this model. So if we believe that RE assumptions, we will want to claw out some is efficiency improvements over the pooled estimator. This should trigger a memory in your brain. What tools do we have for cases like this where OLS is inefficient? (F)GLS and MLE.

Since we're thinking about efficiency, we'll want to be focus on the random component and what it looks like let $e_{it} = \alpha_i + \varepsilon_{it}$ be the combined stochastic component. Then under our above assumptions we can say that

$$\begin{aligned}\mathrm{Var}(e_{it}|\mathbf{x}_i) &= \mathrm{Var}(\alpha_i|\mathbf{x}_i) + \mathrm{Var}(\varepsilon_{it}|\mathbf{x}_i) = \sigma_\alpha^2 + \sigma_\varepsilon^2 \\ \mathrm{Cov}(e_{it}, e_{it'}) &= \mathrm{Cov}(\alpha_i + \varepsilon_{it}, \alpha_i + \varepsilon_{it'}) = \sigma_\alpha^2\end{aligned}$$

This means that the $T \times T$ covariance matrix Σ for unit i represented by $\mathrm{E}[\varepsilon_i\varepsilon_i'|\mathbf{x}_{it}]$ is dense with $\sigma_\alpha^2 + \sigma_\varepsilon^2$ on the diagonal and σ_α^2 everywhere else. The full $NT \times NT$ covariance matrix Ω is then block diagonal with Σ repeated N times diagonally.

There are several ways to fit random effects model. The first we'll consider is an FGLS approach as it is a little more general. Recall that

$$\hat{\theta}_{\mathrm{FGLS}} = (\mathbf{X}'\widehat{\Omega}^{-1}\mathbf{X})^{-1}\mathbf{X}'\widehat{\Omega}^{-1}y,$$

we can make this a little bit easier on ourselves and our computers by exploiting symmetry.

Because Ω is block diagonal, we can work with just Σ. Already an improvement. We don't really need all of Σ^{-1} either, we really just need its square root in order to pre-treat the data for OLS. As you may recall, there are many different ways to think about the square root of a matrix (e.g., Choleskey). We'll focus on eigenvector decomposition for reasons that hopefully become clear.

Before we get into this, here's a few things you might want to remember about eigenvalues

and eigenvectors.

1. A non-zero vector v_i is an **eigenvector** of a square matrix A if there exists a constant λ_i such that $Av_i = \lambda_i v_i$, where λ_i is called an **eigenvalue**.
2. For an $N \times N$ square matrix there will be N eigenvectors (each of length N) and N eigenvalues. These values may not be unique.
3. Because v_i is non-zero $\det(A - \lambda_i I) = 0$
4. $A = V\Lambda V^{-1}$ where V is a matrix where the ith column is v_i and Λ is a matrix with the corresponding eigenvalues on the diagonal.
5. A and A^{-1} have the same eigenvectors and the eigenvalues of A^{-1} are $1/\lambda$
6. $\det(A) = \prod_i \lambda_i$, and if A is triangular, then $\det(A)$ is also the product of its diagonal elements

So

$$\begin{aligned}\Sigma &= V\Lambda V^{-1} \\ \Sigma^{-1} &= V\Lambda^{-1}V^{-1} \\ \Sigma^{-1/2} &= V[\Lambda^{-1/2}]V^{-1}.\end{aligned}$$

where V is a matrix where each column is an eigenvector and Λ is a matrix with the eigenvalues of Σ on the diagonal. As such the diagonal of $\Lambda^{-1/2}$ is $1/\sqrt{\lambda}$ where λ are the eigenvalues of Σ.

We could compute these each time, but maybe there's a more general solution? Recall that to find the eigenvalues of a matrix we need to solve

$$\det(\Sigma - \lambda I) = 0$$

for all possible values of λ. What do we know about this matrix?

$$\Sigma - \lambda I = I(\sigma_\alpha^2 + \sigma_\varepsilon^2 - \lambda) + \mathbf{1}\mathbf{1}'\sigma_\alpha^2.$$

With $T-2$ steps of Gaussian elimination you can get an upper diagonal matrix with diagonals:

$$(T\sigma_\alpha^2 + \sigma_\varepsilon^2 - \lambda, \sigma_\varepsilon^2 - \lambda, \ldots, \sigma_\varepsilon^2 - \lambda).$$

So the determinant of this matrix is the product of these diagonals and the eigenvalues are the values of λ that make this product 0. So what are the eigenvalues?

- 1 is $T\sigma_\alpha^2 + \sigma_\varepsilon^2$
- The other $T - 1$ are σ_ε^2

Ok, remember the goal is to make an easy-to-use form of $\Sigma^{-1/2}$ that won't be dependent

on sample size, so what's next? We have the eigenvalues for Σ, now we need them for Σ^{-1}. Thankfully that's as easy as

$$1/\lambda = \left(\frac{1}{T\sigma_\alpha^2 + \sigma_\varepsilon^2}, \frac{1}{\sigma_\varepsilon^2}, \dots \frac{1}{\sigma_\varepsilon^2} \right).$$

This gives us the Λ matrix. Now we need eigenvectors, Working with Σ, let's start with the $T-1$ values that are σ_ε^2. Consider an arbitrary row t from Σ, we need it to solve (from property 1 above):

$$v_t\sigma_\alpha^2 + v_t\sigma_\varepsilon^2 + \sum_{s \neq t} v_s\sigma_\alpha^2 = \sigma_\varepsilon^2 v_t.$$

Note that σ_ε^2 appears only once on the RHS, so v_t for sure needs to be non-zero. This however means that $v_t\sigma_\alpha^2$ sticks around, so we need to cancel it out with something from the sum. The easiest way? For eigenvectors 2 through T let $v_t = 1$, $v_1 = -1$ and everything else be 0. This just leaves eigenvector 1 which solves

$$v_1\sigma_\alpha^2 + v_1\sigma_\varepsilon^2 + \sum_{s=2}^{T} v_s\sigma_\alpha^2 = v_1 T\sigma_\alpha^2 + v_1\sigma_\varepsilon^2.$$

The obvious? $v = 1$. Now we've got it so this gives us

$$V = \begin{bmatrix} 1 & -\mathbf{1}_{T-1} \\ \mathbf{1}_{T-1} & \mathbf{I}_{T-1} \end{bmatrix}.$$

So we've got all the pieces now for

$$\begin{aligned} \Sigma^{-1/2} &= V[\Lambda^{-1/2}]V^{-1} \\ &= \frac{1}{\sigma_\varepsilon}\left[I_T - \frac{\omega}{T}\mathbf{1}\mathbf{1}' \right] \\ \omega &= 1 - \frac{\sigma_\varepsilon}{\sqrt{T\sigma_\alpha^2 + \sigma_\varepsilon^2}}. \end{aligned}$$

And the FGLS estimates of the RE model become

$$\hat{\theta}_{\text{FGLS}} = (\tilde{\mathbf{X}}'\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}'\tilde{y},$$

where

$$\begin{aligned} \tilde{y}_i &= \Sigma^{-1/2}y_i = (y_i - \hat{\omega}\bar{y}_i)/\sigma_\varepsilon \\ \tilde{\mathbf{x}}_i &= \Sigma^{-1/2}\mathbf{x}_i = (\mathbf{x}_i - \hat{\omega}\bar{\mathbf{x}}_i)/\sigma_\varepsilon. \end{aligned}$$

Fortunately this contains only two parameters σ_α^2 and σ_ε^2, both of which can be estimated using the pooled residuals and the RE assumptions.

Here are the steps:

1. Fit the model using pooled OLS (consistent), call the residuals in this case $\hat{e}$ where

$$\hat{e}_{it} = y_{it} - \hat{\theta}_p' \mathbf{x}_{it}.$$

   Note that the pooled residuals are estimates of e_{it}, and so $\hat{e}'\hat{e}/(NT)$ is a consistent estimator $\text{Var}(e_{it}|\mathbf{x}_i) = \text{Var}(u_i|\mathbf{x}_i) + \text{Var}(\varepsilon_{it}|\mathbf{x}_i) = \sigma_\alpha^2 + \sigma_\varepsilon^2$.

2. Likewise, the pooled residuals can also be used to estimate σ_α^2, how? It's the within-covariance of the residuals

$$\hat{\sigma}_\alpha^2 = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{T(T-1)/2} \sum_{t=2}^{T}\sum_{t'=1}^{t-1} \hat{e}_{it}\hat{e}_{it'},$$

   OR, you can use the within estimator residuals (below) to estimate σ_ε^2 (easier).

3. Use the relationship

$$\sigma_e^2 = \sigma_\alpha^2 + \sigma_\varepsilon^2$$

   to back out the remaining quantity of interest.

4. Build $\hat{\omega}$ as described above and fit using OLS on the transformed data.

Property A4 *Under Assumptions A1.R–A5.R, as N increases, the random effects estimator will exist, be consistent for θ, and asymptotically normal. It is also unbiased and (weakly) more efficient than pooled OLS in finite samples.*

This property follows from ordinary (F)GLS results. The RE estimator will also consistent for θ in T. It will be inconsistent for σ_α^2 in T and this will affect anything we can say about efficiency in the big-T fixed N setting.

Note that for unbalanced panels, you'll need to

1. Be a little more careful estimating σ_α^2 and
2. Estimate ω_i separately for each unit

Both of these changes reflect the varying length T_i within each unit.

This version of the RE model is a semi-parametric way to consider heterogeneity across units. Basically, we assume that conditional on the observables all the remaining heterogeneity is

mean-zero noise that is uncorrelated with the observables. We haven't put a distributional assumption on that noise yet except to say that it each u_i is an exogenous iid shock from an unknown distribution with several moments.

Identification in this case comes from correctly specifying the rest of the model. In this way, it is very similar to a standard cross-sectional linear models. The main identification comes from having no omitted variables in either z_i or x_{it} that are correlated with both the treatment of interest and the outcome of interest.

As such the assumptions needed for the RE model to identify an effect of interest are the same as the pooled model. What we've done is say, "look we recognize that there could be heterogeneity across units. We're going to model that heterogeneity in way that pooled OLS is consistent but inefficient. We can gain that efficiency back using the FGLS approach." Is that worth much? It's not nothing, but it very much relies on thinking the RE assumptions are reasonable.

It's worth asking the question at this point, what do to with the standard errors? Under the RE assumptions we've made so far, the standard GLS variance matrix,

$$\text{Var}(\hat{\theta}_{RE}|\mathbf{X}) = \sigma^2_{FGLS}(\tilde{\mathbf{X}}'\tilde{\mathbf{X}})^{-1},$$

is correct, but recall that these assumptions are fairly strong (i.e., iid observations once we condition on u_i and two levels of homoskedasticity). Note that using the transformation above $\sigma^2_{FGLS} = 1$, some softwares do different transformations, so be careful. Often we are not so convinced of all our RE assumptions, but if we think they're mostly reasonable we may want to consider a clustered covariance matrix with the FGLS estimates. This is perhaps controversial as we make the efficiency claims largely on the basis of these assumptions, but then say we're not so sure about them if we cluster. The extent to which the clustered standard errors differ from the GLS standard errors may tell us something about how believable the RE assumptions are (e.g., King and Roberts).

Now two more points before we move on. First, this is not the only way to fit this model. Perhaps more common is a maximium likelihood approach that requires additional parametric assumptions

Assumption A6.R The stochastic components are normally distributed $u_i \sim N(0, \sigma^2_\alpha)$, $\varepsilon_{it} \sim N(0, \sigma^2_\varepsilon)$.

this parametric addition means we don't have to do a multi-step approach. Under this

assumption

$$y_i|\mathbf{x}_i \sim N\left(\left[\alpha + \beta' x_{it} + \gamma' z_i\right]_{t=1}^{T_i}, \Sigma_i\right),$$

which suggests a straightforward log-likelihood function where each unit is a draw from this multivariate normal, giving us

$$L(\theta|y) = \sum_{i=1}^{N} -\frac{1}{2}\log(\det(\Sigma_i)) - \frac{1}{2}(y_i - \theta'\mathbf{x}_i)'\Sigma_i^{-1}(y_i - \theta'\mathbf{x}_i),$$

which we can simplify a bit further using what we know about the eigenvector decomposition of Σ_i, in that

$$\begin{aligned}
\det(\Sigma_i) &= (T_i\sigma_\alpha^2 + \sigma_\varepsilon^2)(\sigma_\varepsilon^2)^{T_i-1} \\
\frac{1}{2}\log(\det(\Sigma_i)) &= \frac{1}{2}\log(T_i\sigma_\alpha^2 + \sigma_\varepsilon^2) + \frac{T_i - 1}{2}\log(\sigma_\varepsilon^2) \\
\frac{1}{2}(y_i - \theta'\mathbf{x}_i)'\Sigma_i^{-1}(y_i - \theta'\mathbf{x}_i) &= \frac{1}{2}[(e_i - \omega_i\bar{e}_i)/\sigma_\varepsilon]'[(e_i - \omega_i\bar{e}_i)/\sigma_\varepsilon]
\end{aligned}$$

Second, this approach to modeling heterogeneity is unlikely to satisfy many people because the exogeneity assumption is quite strong. Likewise, the specific assumptions required for the RE estimator to be more efficient than the pooled estimator requires both within-unit independence and homoskedasticity at both the observation and unit levels. As such, we will set this framework aside for a moment and consider another approach to modelling unobserved heterogeneity.

1.2 The fixed effects model

All right, so how might we think about unobserved heterogeneity? Again, we start from **Assumption A1**

$$y_{it} = \alpha_i + \beta' x_{it} + \gamma' z_i + \varepsilon_{it}.$$

This time we change it to be **Assumption A1.F**

$$y_{it} = \alpha_i + \beta' x_{it} + \varepsilon_{it}$$

where

$$\alpha_i = \alpha + \gamma' z_i + u_i$$

Unlike the RE model, u_i are fixed parameters not draws from a random variable (thus the names) and contain everything unobserved about unit i that is time-invariant. By estimating

this fixed, overall constant for each unit, we tuck $\gamma' z_i$ into α_i along with everything that is time-invariant. ote that this constant controls for **all** time-invariant heterogeneity, even things we didn't think of or can't measure. So we lose identification of γ, but we gain insulation from a range of omitted variables. In this way, the fixed effects model is a very important tool for fighting endogeneity as it eliminates any concerns about omitted variable bias from time-invariant sources. We will also return to iid units rather than observations (Assumption A2) and maintain strict exogeneity within units (Assumption A3).

This leaves us with a very similar setup to the pooled model. However, we haven't said anything about the correlation between u_i and the exogeneity of x_{it}. In the pooled and RE settings we assumed that any within-unit deviations from the overall constant α could be safely ignored by either a) knowing/including it z_i, b) leaving it outside the model as either either part of ε_{it} (pooled) or the random u_i (RE). Now however, we're going to ask, when is that assumption reasonable?

Suppose we fit the model in Assumption A1.F using either a pooled or RE estimator. In this case we include any observed z_i, but are leaving the unobserved u_i in the error term, as we've done before. This leaves us with a joint error term $e_{it} = \varepsilon_{it} + u_i$, such that

$$\mathrm{E}[\hat{\theta}_p|\mathbf{x}_{it}] = \theta + \left(\frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T}\mathbf{x}_{it}\mathbf{x}_{it}'\right)^{-1}\frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T}\mathbf{x}_{it}\,\mathrm{E}[u_i|x_{it}],$$

which is our familiar omitted variable bias result. If there is any correlation between the variables in $\mathbf{x}_{it}$ and the unit-specific heterogeneity u_i, then the pooled estimator (and by extension the random effects estimator) are biased and inconsistent because of this endogeneity.

This poses a notable issue. What can we do to consider a non-parametric form of heterogeneity? Obviously, if we feel ok assuming that it is unrelated to either the treatment or outcome of interest then we're fine to return to the pooled estimator. However, in cases where that's unlikely to be true, we still have some options.

1.2.1 Within-estimator

The first approach we'll consider involves what's known as the "within transformation," using

$$M_i := \mathbf{I}_i - \mathbf{1}_i(\mathbf{1}_i'\mathbf{1}_i)^{-1}\mathbf{1}_i'.$$

Here M_i is the "demeaning" matrix. It's not insulting, but it does subtract the unit-specific mean of any matrices it meets such that

$$\begin{aligned} M_i y_i &= y_i - \bar{y}_i \\ M_i X_i &= \begin{bmatrix} X_{i1} - \bar{X}_{i1} & \dots & X_{iK} - \bar{X}_{iK} \end{bmatrix}. \end{aligned}$$

Let's consider this demeaning approach, such that

$$\begin{aligned} y_{it} - \bar{y}_i &= \beta'(x_{it} - \bar{x}_i) + (\alpha_i - \bar{\alpha}_i) + (\varepsilon_{it} - \bar{\varepsilon}_i) \\ y_{it} - \bar{y}_i &= \beta'(x_{it} - \bar{x}_i) + (\varepsilon_{it} - \bar{\varepsilon}_i) \\ M_i y_i &= M_i X_i \beta + M_i \varepsilon_i \end{aligned}$$

We can fit this model using OLS. Doing so is called the **within estimator**.

Note that because we are using OLS to fit the within model. We inherit all the good properties of OLS is so the within estimator is unbiased and consistent, if not efficient under Assumptions A1.F, A2, and A3. With appropriate rank conditions, we can also say that the correct variance matrix is the clustered variance matrix using the within transformed data.

Of additional note, supposed we have homoskedasticity and within-unit independence. From what we know about OLS, this gives us

$$\text{Var}(\hat{\beta}^0_w | X_i) = \sigma^2_\varepsilon \left(\sum_{i=1}^{N} X_i' M_i X_i \right)^{-1},$$

which as you'll show in a problem set is weakly greater than the variance of the pooled OLS estimator

$$\text{Var}(\hat{\beta}^0_p | X_i) = \sigma^2_\varepsilon \left(\sum_{i=1}^{N} X_i' X_i \right)^{-1}.$$

The intuition behind this is that the demeaning process removes some information from each x variable (specifically the cross-section information).

What does this mean for us? Two things. First, it means that we have made a firm choice regarding what information matters. We are only interested in the within-unit variation. This is reflected in the fact that most software packages will report two (or three) different

R^2 values for within estimation

$$R^2 = 1 - \frac{SSR}{SST} = 1 - \frac{\hat{\varepsilon}'\hat{\varepsilon}}{(y - \bar{y})'(y - \bar{y})}$$
$$R^2_{\text{adj}} = 1 - (1 - R^2)\frac{NT - 1}{NT - N - k}$$
$$R^2_{\text{within}} = 1 - \frac{\hat{\varepsilon}'\hat{\varepsilon}}{\sum(y_i - \bar{y}_i)'(y_i - \bar{y}_i)}$$

The differences between the overall R^2 and the within R^2 are that the former tells us how much of the total (cross-sectional and within) variance in y is explained by X plus the unit-specific heterogeneity. The latter tells us just how much of the within-unit changes in y variance is explained by X. The overall R^2 tends to be a lot higher as unit-specific heterogeneity tends to explain a lot.

Second, in this context it tells us that when we have panel data with unobserved heterogeneity and homoskedastic and independent errors, there is a bias-variance trade off. Ignoring the heterogeneity by estimating the pooling model will result in lower variance but more bias. Being robust to the heterogeneity by fitting the fixed-effects model using the within transformation will decrease bias but at the cost of variance. Generally, we're more concerned with bias in estimating treatment effects, but it's worth remembering that we don't get it for free.

1.2.2 Dummy variable estimator

Another way may be to just estimate the time invariant parameters directly for each unit. Let

$$\alpha_i = \alpha + \gamma' z_i + u_i,$$

then the model becomes

$$y_{it} = \beta' x_{it} + \alpha_i + \varepsilon_{it},$$

which can be fit using OLS with a dummy variable for each unit.

Let $\theta_{LSDV} = (\beta, \alpha_i)_{i=1}^N$ and redefine $\mathbf{X} = [X\ D]$ where D (no subscript) is a $NT \times N$ matrix of dummy variables where each column denotes if the observation is associated with unit i. Notice that we no longer have an overall constant, instead we have a constant for each unit.

This approach is identical to the within transformation such that $\hat{\beta}_w = \hat{\beta}_{LSDV}$. The within estimator saves us from estimating the N unit-specific parameters, which can be quite handy for larger N, but it does not direct estimate the constants. However, this drawback rarely

matters to us in practice, at least for the linear model.

We will consider this equivalence in two steps. First, consider a model with no covariates:

$$y_{it} = \alpha_i + \varepsilon_{it}.$$

What would the least squares estimate be for α_i? The sample mean for group i (i.e, $\hat{\alpha}_i = \bar{y}_i$). This means that the residuals of $\hat{\varepsilon}_{it} = y_{it} - \bar{y}_i$, which of course is the residual vector from the within transformation.

Second, consider the model from Assumption A1.F

$$y_{it} = \alpha_i + \beta' x_{it} + \varepsilon_{it}.$$

In matrix form we can write this as

$$y = X\beta + D\alpha_i + \varepsilon,$$

Note that we can use the Firsch-Waugh-Lovell (FWL) theorem to consider the LSDV estimator of β separately from the LSDV estimator of α. First, let us remind ourselves what the FWL says,

Theorem 6 Firsch-Waugh-Lovell (FWL) *For a regression model of $y = X_1\beta_1 + X_2\beta_2 + \varepsilon$, with $\beta = (\beta_1, \beta_2)$, the OLS estimator of β_2 can be computed using the following algorithm:*

1. *Regress y on X_1 and save the residuals $\hat{e}_1$*
2. *Regress X_2 on X_1 and save the residuals $\hat{e}_2$*
3. *Regress $\hat{e}_1$ on $\hat{e}_2$ to get $\hat{\beta}_2$ and $\hat{\varepsilon}$*

In our case this means that we can compute $\hat{\beta}_{LSDV}$ in the following way

1. Regress y on D and save the residuals $\hat{e}_1$
2. Regress X on D and save the residuals $\hat{e}_2$
3. Regress $\hat{e}_1$ on $\hat{e}_2$ to get $\hat{\beta}_{LSDV}$ and $\hat{\varepsilon}$

Regressing anything on just D, as we showed above, **is** the within transformation. So steps 1 and 2 here are just conducting the within-transformation and step 3 is the within estimator.

The big deal here, is that the with the within/LSDV estimator, the estimated values of β are completely invariant to the values of the fixed effects α_i. This means that we do not need to

put additional assumptions on α_i like we did with the pooled or random effects estimators. The composition of α_i can be correlated with the error terms and it doesn't matter.

As an additional note, when using degree of freedom corrections (or otherwise accounting for degrees of freedom), the correct number includes the N α_i terms even when using the within transformation. Why? Answer: The within transformation involves estimating $N(k+1)$ sample means, but these sample means are themselves directly related to the the N unit-specific intercepts such that

$$\hat{\alpha}_i = \bar{y}_i - \hat{\beta}'_w \bar{x}_i$$

, as such we are still using $N + k$ degrees of freedom even when we don't actually directly estimate the intercepts.

The LSDV/within approach also has a connection to the RE approach. Remember that we used the following weights for the RE estimator

$$\omega_i = 1 - \frac{\sigma_\varepsilon}{\sqrt{T_i \sigma_\alpha^2 + \sigma_\varepsilon^2}}.$$

and the RE-FGLS estimator was OLS applied to the transformed data $(y_i - \hat{\omega}\bar{y})$ and likewise for $\mathbf{x}$. A couple things to note:

1. If $\hat{\sigma}_\alpha^2 = 0$ then $\hat{\omega} = 0$ and $\hat{\theta}_{RE} = \hat{\theta}_p$
2. If $\hat{\sigma}_\varepsilon^2 >> T_i \hat{\sigma}_\alpha^2$, then $\hat{\omega} \approx 0$ and $\hat{\theta}_{RE} \approx \hat{\theta}_p$
3. If $T_i \hat{\sigma}_\alpha^2 >> \hat{\sigma}_\varepsilon^2$, then $\hat{\omega} \approx 1$ and $\hat{\beta}_{RE} \approx \hat{\beta}_w$.

Now note that β in this model reflects only the average within unit changes (thus the name). Basically, by applying the within transformation, we discard the cross-section information and focus on how the treatment affects each individual. At the other end is the pooled model, which uses both the within and the between unit information. In fact, we can consider what the other extreme might be: the **between** estimator

$$\hat{\theta}_{\text{btwn}} = \left(\sum_{i=1}^{N} \bar{\mathbf{x}}_i \bar{\mathbf{x}}_i' \right)^{-1} \sum_{i=1}^{N} \bar{\mathbf{x}} \bar{y}_i.$$

There is little practical use for the between estimator, but it does show us the extreme case where all we care about is the cross-sectional variation and where we care nothing about the within-unit variance. However, we can note the relationships between the pooled, between,

and within estimators such that

$$\begin{aligned}
\hat{\theta}_p &= T_{XX}^{-1} T_{Xy} \\
\hat{\theta}_w &= W_{XX}^{-1} W_{Xy} \\
\hat{\theta}_b &= B_{XX}^{-1} B_{Xy} \\
T_{XX} &= \sum_{i=1}^{N} \mathbf{x}_i \mathbf{x}_i' \\
B_{XX} &= \sum_{i=1}^{N} \bar{\mathbf{x}}_i \bar{\mathbf{x}}_i' \\
W_{XX} &= T_{XX} - B_{XX} \\
\hat{\theta}_{RE} &= (W_{XX} + \lambda B_{XX})^{-1} (W_{Xy} + \lambda B_{Xy}) \\
\lambda &= (1 - \omega)^2.
\end{aligned}$$

All of this to say that the random effects estimator can also be expressed as a weighted combination of the between and within estimators. When they're weighted equally ($\lambda = 1$) then we have the pooled estimator, as λ moves to 0 (favoring the within variance), we get the within estimator.

1.2.3 First differences

Yet another way to consider the FE model is to remove the heterogeneity by subtracting y_{t-1} from both sides

$$\begin{aligned}
y_{it} - y_{t-1} &= (\alpha_i - \alpha_i) + \beta'(x_{it} - x_{it-1}) + \varepsilon_{it} - \varepsilon_{it-1} \\
\Delta y_{it} &= \beta' \Delta x_{it} + \Delta \varepsilon_{it}.
\end{aligned}$$

All the time-invariant heterogeneity is removed and OLS becomes a good estimator for β.

In matrix form this looks like

$$\begin{aligned}
\Delta_i y_i &= \Delta_i X_i + \Delta_i \varepsilon_i \\
\Delta_i &= \underbrace{\begin{bmatrix} -1 & 1 & 0 & \dots & 0 & 0 \\ 0 & -1 & 1 & & 0 & 0 \\ \vdots & & & \ddots & & \vdots \\ 0 & 0 & 0 & \dots & -1 & 1 \end{bmatrix}}_{T_i - 1 \times T_i} \\
\hat{\beta}_{FD} &= \left(\sum_{i=1}^{N} X_i' \Delta_i' \Delta_i X_i \right)^{-1} \left(\sum_{i=1}^{N} X_i' \Delta_i' \Delta_i y_i \right)
\end{aligned}$$

Because this is simply pooled OLS on the differenced data, we can obtain a few properties with ease:

Property A5 *Under Assumptions A1.F, A2, and A3 the first differences estimator $\hat{\beta}_{FD}$ is unbiased for β.*

With additional rank and moment assumptions, we can also obtain

Property A6 *$\hat{\beta}_{FD}$ is consistent in N and asymptotically normal with the clustered variance matrix on the differenced data.*

Note that when $T = 2$ the FD estimator will be identical to the within estimator, but this does not hold for $T > 2$. Additionally, if we wanted to make some independence and homoskedasticity assumptions on the undifferenced errors ε_{it} then we can note that the differenced errors are correlated within units

$$\text{Var}(\Delta_i \varepsilon_i | X_i) = \Delta_i \Delta_i' \sigma_\varepsilon^2,$$

where $\Delta_i \Delta_i'$ is a matrix with 2 on the diagonal, -1 on the first off-diagonals, and 0 everywhere else.

This means that we can eek out some improvements via GLS since we know the variance, such that

$$\hat{\beta}_{FD}^{GLS} = \left(\sum_{i=1}^{N} X_i' \Delta_i' (\Delta_i \Delta_i')^{-1} \Delta_i X_i \right)^{-1} \left(\sum_{i=1}^{N} X_i' \Delta_i' (\Delta_i \Delta_i')^{-1} \Delta_i y_i \right)$$

After some algebra, the inner parts work out to be

$$\Delta_i' (\Delta_i \Delta_i')^{-1} \Delta_i = \mathbf{I}_i - \mathbf{1}_i (\mathbf{1}_i' \mathbf{1}_i)^{-1} \mathbf{1}_i' := M_i.$$

This is the demeaning matrix again. So we can rewrite the GLS-FD estimator as

$$\hat{\beta}_{FD}^{GLS} = \left(\sum_{i=1}^{N} X_i' M_i X_i \right)^{-1} \left(\sum_{i=1}^{N} X_i' M_i y_i \right).$$

This is the within estimator! So if the errors are iid and homoskedastic then the within estimator is BLUE by the GLS properties. However, that's probably not going to be something we want to lean on very often, but it's 1 point in favor of within over FD. Of course, if $\Delta \varepsilon_{it}$ are iid and homoskedastic then FD is BLUE. More generally, the LSDV/within estimators will only be identical to the FD estimates when $T = 2$.

1.3 Model testing and comparisons

The next thing you should want to know is when do you want to use the pooled, or random effects FGLS, or the LSDV/within. As we've mentioned, the fixed effects estimators will be consistent in the widest set of cases, however, this can come at some efficiency losses. Likewise, in some cases, we now know that the decision may not matter too much (i.e., as T increases the differences between fixed and random effects will be less pronounced, all else equal). Table 1.1 outlines some of the important differences among the models and estimators we've discussed so far.

Okay, so now you're thinking I don't want inconsistent estimates, but efficiency is nice. How do I choose among these estimator?

As we mentioned, even if the random effects assumptions are good, the RE-FGLS estimator converges to the within-estimator (below) as T increases. So the efficiency gains are fleeting as T increases while the risk of bias and inconsistency remain. Recall that we can consider the closeness between the two estimators by just looking at the RE weights ω_i.

$$\omega_i = 1 - \frac{\sigma_\varepsilon}{\sqrt{\sigma_\varepsilon^2 + T_i\sigma_\alpha^2}}.$$

The closer ω_i is to 1 the more similar the estimates are, the fewer efficiency gains if the RE assumptions are correct. Likewise, if the RE assumptions are not satisfied the estimator is inconsistent.

However, in most cases there will be differences. In these cases, we have should have good reasons for choosing the estimators we do. Most of the time, we care about consistency more than efficiency. This is a good reason to make the within/LSDV your first choice (or the "mostly harmless" choice).

If you're not yet convinced that random effects are mostly meh, or you really think there's a good reason for that approach, you can consider two different types of hypothesis tests. The first is the common textbook text for this question: The Hausman test. The Hausman test should be considered for when you think random effects are right, and you want to provide evidence in support of that decision. It should *not* be used to make a selection when you're agnostic about fixed versus random effects. If you're agnostic, use the fixed effects because they require fewer assumptions.

The null hypothesis is that $\hat{\beta}_0$ and $\hat{\beta}_1$ are both consistent. The alternative hypothesis is that only $\hat{\beta}_1$ is consistent.. To put this another way, the null is that $q = \hat{\beta}_1 - \hat{\beta}_0 \xrightarrow{p} 0$. Note this is

Table 1.1: Comparing panel estimators

	Pooled	RE-FGLS	FD	LSDV/within
Pooled model: $y_{it} = \alpha + \beta' x_{it} + \gamma' z_i + \varepsilon_{it}$ $(\mathbf{x}_i, \varepsilon_i)$ iid $\mathrm{E}[\varepsilon_{it}\|\mathbf{x}_i] = 0$	• Unbiased and consistent in N. In T, if data are stationary and ergodic. • BLUE if ε_{it} are iid homoskedastic. • Asymptotically efficient if ε_{it} are iid normal and homoskedastic.	• Unbiased and consistent in N for θ. In T, if data are stationary and ergodic. • No efficiency gains over the pooled estimator. • RE covariance matrix may be incorrect b/c of within-unit iid assumptions	• Unbiased and consistent in N for β. • In T if data are stationary and ergodic	• Unbiased and consistent in N for β. • In T if data are stationary and ergodic
RE model: $y_{it} = \alpha + \beta' x_{it} + \gamma' z_i$ $+\alpha_i + \varepsilon_{it}$ $\mathrm{E}[\alpha_i] = 0$ $\mathrm{Cov}(\mathbf{x}_{it}, \alpha_i) = 0$ $\mathrm{Cov}(\varepsilon_{it}, \alpha_i) = 0.$ $(\mathbf{x}_{it}, \varepsilon_{it})$ iid $\mathrm{E}[\varepsilon_{it}\|\mathbf{x}_i] = 0$ $\mathrm{E}[\varepsilon_{it}^2\|\mathbf{x}_i] = \sigma_\varepsilon^2$ $\mathrm{E}[\alpha_i^2\|\mathbf{x}_i] = \sigma_\alpha^2$	Unbiased and consistent N . In T if data are stationary and ergodic.	• Unbiased and consistent in N. In T if data are stationary and ergodic. • BLUE • Asymptotically efficient if α_i and ε_{it} are normal	See above	See above
FE model: $y_{it} = \beta' x_{it} + \alpha_i + u_{it}$ (x_i, ε_i) iid $\mathrm{E}[\varepsilon_{it}\|\mathbf{x}_i] = 0$	Biased and inconsistent	Biased and inconsistent in N. Consistent in T as $\omega \to 1$, if data are stationary and ergodic.	See above. BLUE if $\Delta\varepsilon_{it}$ are iid and homoskedastic	See above. BLUE if ε_{it} iid and homoskedastic

a slightly different kind of hypothesis than we're used to, because it relates to the limiting value of an estimate rather than whether the true parameters are a particularly value.

Hausman derives this hypothesis test for the case where $\hat{\beta}_0$ is the asymptotically efficient estimator of β (i.e., RE v FE). In this case we can consider the (joint) distribution of the estimators. We know that the sampling distributions are individually normal (asymptotically), and we will add the additional assumption that they are jointly normal.

So now we need to know the distribution of q, we start with the joint distribution:

$$\sqrt{N}\begin{bmatrix}\hat{\beta}_1 - \beta \\ \hat{\beta}_0 - \beta\end{bmatrix} \overset{asy}{\sim} N\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}\sigma_\varepsilon^2 \operatorname{E}[X'MX] & N\operatorname{Cov}(\hat{\beta}_1, \hat{\beta}_0) \\ N\operatorname{Cov}(\hat{\beta}_1, \hat{\beta}_0) & \operatorname{E}[X'\Sigma^{-1}X]\end{bmatrix}\right)$$

and this means that $q = \hat{\beta}_1 - \hat{\beta}_0$ is also asymptotically normal with mean 0 and variance

$$\operatorname{Var}(q) = \operatorname{Var}(\hat{\beta}_1) + \operatorname{Var}(\hat{\beta}_0) - 2\operatorname{Cov}(\hat{\beta}_1, \hat{\beta}_0),$$

under the null. Now we don't typically know the covariance across estimators so $\operatorname{Cov}(\hat{\beta}_1, \hat{\beta}_0)$ is unclear (although we could—and maybe should—bootstrap it). We need to be clever. Let's rewrite q such that we get

$$\begin{aligned}\hat{\beta}_1 &= \hat{\beta}_0 + q \\ \operatorname{Var}(\hat{\beta}_1) &= \operatorname{Var}(\hat{\beta}_0) + \operatorname{Var}(q) + 2\operatorname{Cov}(\hat{\beta}_0, q).\end{aligned}$$

Claim: $\operatorname{Cov}(\hat{\beta}_0, q) = 0$

Proof. Suppose not, that is, let $\operatorname{Cov}(\hat{\beta}_0, q) \neq 0$. We can define another estimator $\hat{\beta}_2 = \hat{\beta}_0 + rAq$, where A is an arbitrary matrix and r an arbitrary scalar. Because $q \xrightarrow{p} 0$, we know that $\hat{\beta}_2$ is consistent and asymptotically normal with variance

$$\operatorname{Var}(\hat{\beta}_2) = \operatorname{Var}(\hat{\beta}_0) + rA\operatorname{Cov}(\hat{\beta}_0, q) + r\operatorname{Cov}(\hat{\beta}_0, q)A' + r^2 A\operatorname{Var}(q)A'.$$

Let

$$f(r) = \operatorname{Var}(\hat{\beta}_2) - \operatorname{Var}(\hat{\beta}_0) = 2rA\operatorname{Cov}(\hat{\beta}_0, q)A' + r^2 A\operatorname{Var}(q)A' \geq 0$$

be the difference in variances between this new estimator and the efficient estimator $\hat{\beta}_0$. The derivative of f derivative wrt to r is

$$D_r f(r) = A\operatorname{Cov}(\hat{\beta}_0, q) + \operatorname{Cov}(\hat{\beta}_0, q)A' + 2rA\operatorname{Var}(q)A'.$$

Now consider the special case where $r = 0$ and $A = -\operatorname{Cov}(\hat{\beta}_0, q)$

$$D_r f(0) = -2 \operatorname{Cov}(\hat{\beta}_0, q)' \operatorname{Cov}(\hat{\beta}_0, q).$$

If $\operatorname{Cov}(\hat{\beta}_0, q) \neq 0$, then this is a quadratic times a negative constant. As such, $D_r f(0) < 0$, which is to say that $f(r)$ is decreasing in r at $r = 0$.

But, note that $f(0) = 0$, so for some small $r > 0$, $f(r)$ will be negative. However, this contradicts the the fact that $\hat{\beta}_0$ is the efficient estimator. Therefore, we conclude that $\operatorname{Cov}(\hat{\beta}_0, q) = 0$. □

This tells us two identical things:

1. The variance of q
$$\begin{aligned} \operatorname{Var}(\hat{\beta}_1) &= \operatorname{Var}(\hat{\beta}_0) + \operatorname{Var}(q) + 0 \\ \operatorname{Var}(q) &= \operatorname{Var}(\hat{\beta}_1) - \operatorname{Var}(\hat{\beta}_0), \end{aligned}$$

2. The actual covariance of $\hat{\beta}_1$ and $\hat{\beta}_0$

$$\begin{aligned} \operatorname{Cov}(\hat{\beta}_1 - \hat{\beta}_0, \hat{\beta}_0) &= 0 \\ \operatorname{Cov}(\hat{\beta}_1, \hat{\beta}_0) - \operatorname{Cov}(\hat{\beta}_0, \hat{\beta}_0) &= 0 \quad \text{Properties of covariance} \\ \operatorname{Cov}(\hat{\beta}_1, \hat{\beta}_0) &= \operatorname{Var}(\hat{\beta}_0) \end{aligned}$$

Returning to the test statistic, we can now construct a standard χ^2 test based on q such that

$$q' \left(\operatorname{Var}(\hat{\beta}_1) - \operatorname{Var}(\hat{\beta}_0) \right)^{-1} q \overset{asy}{\sim} \chi^2_k,$$

where k is the length of β.

Note, that the RE estimator is only more efficient under the RE assumptions, which include homoskedasticity and iid observations. If either of these fails, this test is suspect. As such, we can only use the "classical" variance matrices

$$\begin{aligned} V_1 &= \hat{\sigma}^2_\varepsilon \left[\sum_i X' M_i X \right]^{-1} \\ V_0 &= \mathbf{X}' \widehat{\Omega}^{-1} \mathbf{X} \qquad \text{restricted to } \beta. \end{aligned}$$

Because of this test's reliance on within-unit independence and homoskedasticity, we may want to consider alternatives. One that you'll think about in a problem set or something will consider the power and size of a version based on a clustered bootstrap.

Now because we want to know if the iid assumptions have any bite, we'll want to know if there is any autocorrelation in the residuals. Wooldridge recommends a panel version of the standard Bruesch-Godfrey test that simply regresses $\hat{\varepsilon}_{it}$ on $\hat{\varepsilon}_{it-1}$.

1.3.1 CRE

There is another approach though which can accommodate more interesting covariance structures without concern. This is based on work by Mundlak, who gifted us *another* estimator for the linear fixed effects model that is also equivalent to the LSDV/within called the **correlated random effects** estimator. Here we adjust the fixed effects model such that

$$\begin{aligned}
y_{it} &= \alpha_i + \beta' x_{it} + \varepsilon_{it} \\
\alpha_i &= \alpha + \gamma' \bar{x}_i + u_i \\
u_i &\sim f(0, \sigma_\alpha^2) \\
\mathbf{x}_{it} &= \begin{bmatrix} 1 & x_{it} & \bar{x}_i \end{bmatrix} \\
\mathrm{E}[u_i | \mathbf{x}_i] &= \mathrm{E}[u_i \varepsilon_i | \mathbf{x}_i] = 0 \\
\mathrm{Cov}(u_i + \varepsilon_{it}, u_i + \varepsilon_{it'}) &= \sigma_\varepsilon^2 I_T + \sigma_\alpha^2 \mathbf{1}\mathbf{1}'
\end{aligned}$$

What's happening here? Well we're blending the RE and FE models a bit. If the RE assumptions are correct, then we should find that $\gamma = 0$ and then these α_i simplify to the standard random intercept from that approach. However, if $\gamma \neq 0$ then we have incorporated a way for the observe covariates x to be correlated with the unobserved heterogeneity α_i. Note that we are retaining iid within-unit observations here (from the RE setup), so the only within-unit autocorrelation is in the form of the constant u_i.

Essentially, we are accommodating the unobserved heterogeneity by modeling it's relationship to the observables and controlling for that. We are directly controlling for deviations from the within means (as in the within model), while deviations of y_{it} from $\bar{y}_i$ and u_{it} from u_i are captured in α_i and α. And indeed $\hat{\beta}_{CRE} = \hat{\beta}_{LSDV} = \hat{\beta}_w$. To see this consider the following

alternative derivation starting with the within model:

$$\begin{aligned}
y_{it} - \bar{y}_i &= \beta_w'(x_{it} - \bar{x}_i) + (\varepsilon_{it} - \bar{\varepsilon}_i) \\
y_{it} &= \beta_w'(x_{it} - \bar{x}_i) + (\varepsilon_{it} - \bar{\varepsilon}_i) + \bar{y}_i \\
\bar{y}_i &= \alpha + u_i + \beta_b'\bar{x}_i + \bar{\varepsilon}_i \\
y_{it} &= \beta_w'(x_{it} - \bar{x}_i) + (\varepsilon_{it} - \bar{\varepsilon}_i) + \alpha + u_i + \beta_b'\bar{x}_i + \bar{\varepsilon}_i \\
&= \beta_w' x_{it} + (\beta_w - \beta_b)'\bar{x}_i + \alpha + u_i + \varepsilon_{it} \\
&= \beta_w' x_{it} + \gamma'\bar{x}_i + \alpha + u_i + \varepsilon_{it}
\end{aligned}$$

Which is to say that we get the within estimates back on x_{it} and the difference between the within and between estimates back on $\bar{x}_i$. This gives us another nice alternative to dummy variables that uses fewer parameters.

Now it also suggests a specification test. Namely if $\alpha = 0$ (i.e., $\beta_w = \beta_b$), then the there are no real differences between the within and between estimators and we can use the more efficient RE estimator (or the pooled estimator if we still want to avoid questionable RE assumptions). This becomes an ordinary Wald test of the hypothesis that $\gamma = 0$. Unlike the Hausman test, the Wald test is well defined and applicable with (cluster) robust covariance matrices.

Note that u_i can be safely ignored here regardless of whether it is correlated with the observables or not because we have obtained the within estimates on the observables. As such we can treat it as either fixed or random. We can leave it in the error term (i.e., fit the above equation with the pooled estimator) or treat it as random and fit the GLS. Regardless, we'll get the within estimates for β.

1.4 Application

In this example we're going to be working with data from Choulis, Escribá-Folch, and Mehri (2024, *JOP*).[1] In this paper, they consider how the presence of secret police within a country affect anti-regime protests.

The outcome of interest is a latent measure based on combining information from several different protest datasets. The treatment of interest is whether there is a secret police organization within that country-year observation (binary). They also consider several control variables include population, GDP per captia, economic growth, politically exclude ethnic groups, protests in neighboring countries, civil conflict, and coup attempts. We will take

[1] https://doi.org/10.1086/729953

their specification at face value and observe the following specification

$$\text{Protests}_{it} = \alpha_i + \beta_1 \text{Secret Police}_{it} + x_{it}'\gamma + \varepsilon_{it}.$$

We will consider pooling, random effects, and FE estimation.

```
## data manipulation packages
library(readstata13)
library(data.table)

## econometrics packages
library(lmtest)
library(car)
library(sandwich)
library(fixest)
library(lme4)
library(clubSandwich)

## tables and figures
library(modelsummary)

## checking out the data
protests <- read.dta13("Rcode/datasets/Replication_secpol_protestComplete.dta")
protests <- data.table(protests)
protests <- protests[order(ccode, year),]

colnames(protests)
```

```
##  [1] "ccode"            "year"             "country"
##  [4] "Region"           "secretpol_revised" "pop"
##  [7] "gdp_pc"           "intrastate"       "polity2"
## [10] "attempt"          "theta_mean"       "physint"
## [13] "disap"            "kill"             "polpris"
## [16] "tort"             "Capacity"         "v2clrspct"
## [19] "v2stfisccap"      "v2terr"           "v2cseeorgs"
## [22] "v2csreprss"       "v2csprtcpt"       "v2csantimv"
## [25] "v2csstruc_1"      "solschdum"        "urbanpop"
```

```
## [28] "l12gr"             "xpers"             "lexclpop"
## [31] "effectivenumber"   "mean3"             "mean5"
## [34] "nbr_mean3"         "nbr_mean5"
```

```
## panel dimenions
length(unique(protests$ccode))
```

```
## [1] 208
```

```
summary(protests[, length(year), by = ccode])
```

```
##      ccode             V1
##  Min.   :  2.0   Min.   : 1.00
##  1st Qu.:313.8   1st Qu.:69.00
##  Median :466.0   Median :69.00
##  Mean   :479.9   Mean   :63.76
##  3rd Qu.:694.5   3rd Qu.:69.00
##  Max.   :990.0   Max.   :69.00
```

```
## adjust the variables based on their replication file

## Normalize the latent varaible to be  mean 0, var 1
protests[, Protest := scale(mean5)]
protests[, nbr_protest := scale(nbr_mean5)]

## create the controls: lag(log(pop)), lag(log(gdp_pc), lag(excluded population))
protests[, `:=` (l.ln_pop = shift(log(pop+1)),
                 l.ln_gdppc = shift(log(gdp_pc)),
                 l.lexclpop = shift(lexclpop)),
         by=ccode]

## model formula
f1 <- Protest~ secretpol_revised + l.ln_pop + l.ln_gdppc  + l12gr+  l.lexclpop+
  nbr_protest+intrastate+attempt


## Fitting with the pooled esetimator
pooled  <- lm(f1, data=protests, x=TRUE)
summary(pooled)
```

```
##
## Call:
## lm(formula = f1, data = protests, x = TRUE)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.18241 -0.48782  0.00395  0.48523  1.98517
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -6.265451   0.185713 -33.737  < 2e-16 ***
## secretpol_revised -0.072502   0.031246  -2.320   0.0204 *
## l.ln_pop           0.337756   0.009246  36.529  < 2e-16 ***
## l.ln_gdppc         0.115105   0.010665  10.793  < 2e-16 ***
## l12gr             -0.011531   0.001964  -5.872 4.73e-09 ***
## l.lexclpop         0.101536   0.043745   2.321   0.0203 *
## nbr_protest        0.158305   0.013073  12.109  < 2e-16 ***
## intrastate         0.192116   0.031781   6.045 1.66e-09 ***
## attempt            0.217309   0.048047   4.523 6.32e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6917 on 3245 degrees of freedom
##   (10009 observations deleted due to missingness)
## Multiple R-squared:  0.4165, Adjusted R-squared:  0.415
## F-statistic: 289.5 on 8 and 3245 DF,  p-value: < 2.2e-16
```

```
## To make life easy
## We're going to restrict ourselves to just the used sample
protests <- protests[as.numeric(row.names(pooled$model)), ]

## Let's consider the residual autocorrelation
## in choosing standard errors
protests[, e.hat := pooled$residuals]
protests[, L.e.hat := shift(e.hat), by =ccode]
```

```r
summary(lm(e.hat~L.e.hat, data=protests)) #that's pretty high!
```

```
##
## Call:
## lm(formula = e.hat ~ L.e.hat, data = protests)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.1104 -0.1078 -0.0032  0.1080  0.7920
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.003252   0.003175   1.024    0.306
## L.e.hat     0.970443   0.004629 209.648   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1779 on 3139 degrees of freedom
##   (113 observations deleted due to missingness)
## Multiple R-squared:  0.9333, Adjusted R-squared:  0.9333
## F-statistic: 4.395e+04 on 1 and 3139 DF,  p-value: < 2.2e-16
```

```r
## Clustering the standard errors
Vcl.pooled <- vcovCL(pooled, cluster=protests$ccode)
round(coeftest(pooled, Vcl.pooled), 5)
```

```
##
## t test of coefficients:
##
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -6.26545    0.72706 -8.6175  < 2e-16 ***
## secretpol_revised -0.07250    0.10565 -0.6863  0.49259
## l.ln_pop           0.33776    0.03442  9.8120  < 2e-16 ***
## l.ln_gdppc         0.11510    0.05135  2.2416  0.02505 *
## l12gr             -0.01153    0.00384 -3.0029  0.00269 **
## l.lexclpop         0.10154    0.15857  0.6403  0.52200
## nbr_protest        0.15830    0.05752  2.7523  0.00595 **
```

```
## intrastate          0.19212    0.08844  2.1722  0.02992 *
## attempt             0.21731    0.07627  2.8493  0.00441 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## Suppose we wanted to bootstrap we have the clustered bootstrap
pooled.boot <- t(replicate(50, {
  idx <- sample(unique(protests$ccode),
                size=length(unique(protests$ccode)),
                replace=TRUE)
  d <- copy(protests)
  d <- d[unlist(sapply(idx, \(x){which(d$ccode==x)}))]
  pooled.bs <- lm(f1, dat=d)
  pooled.bs$coef
}))
round(coeftest(pooled, var(pooled.boot)), 5)
```

```
##
## t test of coefficients:
##
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -6.26545    0.90032 -6.9591  < 2e-16 ***
## secretpol_revised -0.07250    0.14023 -0.5170  0.60519
## l.ln_pop           0.33776    0.04292  7.8701  < 2e-16 ***
## l.ln_gdppc         0.11510    0.06132  1.8772  0.06059 .
## l12gr             -0.01153    0.00398 -2.9005  0.00375 **
## l.lexclpop         0.10154    0.15421  0.6584  0.51031
## nbr_protest        0.15830    0.05551  2.8516  0.00438 **
## intrastate         0.19212    0.08414  2.2832  0.02248 *
## attempt            0.21731    0.07474  2.9075  0.00367 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## And the clustered bayesian bootstrap
pooled.bayes.boot <- t(replicate(50, {
  Ti <- table(protests$ccode)
  d <- copy(protests)
  weight <- rexp(length(unique(protests$ccode)))
```

```
  weight <- weight/sum(weight)
  d$weight <- rep(weight*length(unique(protests$ccode)), Ti)
  lm(f1, dat=d, weights=weight)$coef
}))
round(coeftest(pooled, var(pooled.bayes.boot)), 5)
```

```
##
## t test of coefficients:
##
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -6.26545    0.75865 -8.2586  < 2e-16 ***
## secretpol_revised -0.07250    0.10021 -0.7235  0.46942
## l.ln_pop           0.33776    0.03859  8.7515  < 2e-16 ***
## l.ln_gdppc         0.11510    0.05014  2.2956  0.02176 *
## l12gr             -0.01153    0.00319 -3.6107  0.00031 ***
## l.lexclpop         0.10154    0.15421  0.6584  0.51030
## nbr_protest        0.15830    0.05755  2.7508  0.00598 **
## intrastate         0.19212    0.09509  2.0203  0.04343 *
## attempt            0.21731    0.07011  3.0995  0.00196 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## We can consider fixed effects estimators too. Starting with the LSDV
lsdv  <- lm(update(f1, .~. -1 + factor(ccode)), data=protests)
Vcl.lsdv <- vcovCL(lsdv, cluster=protests$ccode)
round(coeftest(lsdv, Vcl.lsdv)[1:8,], 4)
```

```
##                   Estimate Std. Error t value Pr(>|t|)
## secretpol_revised  -0.2716     0.0926 -2.9334   0.0034
## l.ln_pop            0.6411     0.1077  5.9507   0.0000
## l.ln_gdppc         -0.0180     0.0804 -0.2236   0.8231
## l12gr              -0.0041     0.0026 -1.5970   0.1104
## l.lexclpop         -0.0128     0.1075 -0.1190   0.9053
## nbr_protest         0.1088     0.0664  1.6380   0.1015
## intrastate          0.1851     0.0542  3.4151   0.0006
## attempt             0.1141     0.0432  2.6393   0.0083
```

```
## Within transformation
var.names <- colnames(pooled$model)
protests[,paste0(var.names, ".within"):=lapply(.SD, \(x){x- mean(x)}),
                  by=ccode, .SDcols=var.names ]
fwithin <- paste0(var.names[1], ".within ~ -1 + ",
                  paste0(var.names[-1], ".within", collapse=" + "))
within1 <- lm(fwithin, data=protests)
Vcl.within1 <- vcovCL(within1, cluster=protests$ccode)
round(coeftest(within1, Vcl.within1), 4)
```

```
##
## t test of coefficients:
##
##                           Estimate Std. Error t value Pr(>|t|)
## secretpol_revised.within  -0.2716     0.0910 -2.9858   0.0028 **
## l.ln_pop.within            0.6411     0.1058  6.0571   <2e-16 ***
## l.ln_gdppc.within         -0.0180     0.0790 -0.2276   0.8199
## l12gr.within              -0.0041     0.0025 -1.6255   0.1042
## l.lexclpop.within         -0.0128     0.1056 -0.1211   0.9036
## nbr_protest.within         0.1088     0.0653  1.6673   0.0956 .
## intrastate.within          0.1851     0.0533  3.4762   0.0005 ***
## attempt.within             0.1141     0.0425  2.6865   0.0073 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## The fixest is the better way to go here.  It takes
## a formula of the form y~x|heterogeneity. And automatically
## clusters the variance
within2 <- feols(Protest~ secretpol_revised + l.ln_pop + l.ln_gdppc  + l12gr+  l.lexclpo
  nbr_protest+intrastate+attempt|ccode, data=protests)
summary(within2)
```

```
## OLS estimation, Dep. Var.: Protest
## Observations: 3,254
## Fixed-effects: ccode: 113
## Standard-errors: Clustered (ccode)
##                   Estimate Std. Error   t value   Pr(>|t|)
```

```
## secretpol_revised -0.271642   0.090992 -2.985330 3.4792e-03 **
## l.ln_pop           0.641114   0.105861  6.056172 1.9048e-08 ***
## l.ln_gdppc        -0.017976   0.078980 -0.227601 8.2037e-01
## l12gr             -0.004095   0.002520 -1.625256 1.0692e-01
## l.lexclpop        -0.012796   0.105648 -0.121119 9.0381e-01
## nbr_protest        0.108823   0.065281  1.666995 9.8309e-02 .
## intrastate         0.185147   0.053270  3.475647 7.2605e-04 ***
## attempt            0.114058   0.042462  2.686092 8.3295e-03 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## RMSE: 0.422676     Adj. R2: 0.77316
##                  Within R2: 0.222534
```

```
## truely the same
max(abs(lsdv$residuals-within1$residuals))
```

```
## [1] 3.28626e-14
```

```
max(abs(lsdv$residuals-within2$residuals))
```

```
## [1] 3.153033e-14
```

```
## here we can see the difference between the
## total and within r-squared
c(summary(lsdv)$r.sq, summary(within1)$r.sq)
```

```
## [1] 0.7825909 0.2225345
```

```
## why are these different?
## which of these are unbiased estimates? Which are consistent?
c(summary(lsdv)$sigma, summary(within1)$sigma, sqrt(summary(within2)$sigma2))
```

```
## [1] 0.4307607 0.4231964 0.4307607
```

```
## build the weights for RE=GLS
Ti <- table(protests$ccode) #unbalanced panel so each unit has different weight
Ti <- rep(Ti, Ti)
sigma2.eps <- within2$sigma2 #unbiased and consistent
sigma2.a <- mean(pooled$residuals^2) -sigma2.eps
protests$omega.hat <- 1- sqrt(sigma2.eps/(Ti*sigma2.a+sigma2.eps) )
mean(protests$omega.hat) ## fairly similar on this measure
```

```
## [1] 0.8587313
```

```
protests[,paste0(var.names, ".gls"):=lapply(.SD, \(x){x-omega.hat*mean(x)}),
                  by=ccode, .SDcols=var.names ]
protests[,const.gls:=1-omega.hat]
fgls <- paste0(var.names[1], ".gls ~ -1 + const.gls + ",
               paste0(var.names[-1], ".gls", collapse=" + "))

gls <- lm(fgls, data=protests)
summary(gls)
```

```
##
## Call:
## lm(formula = fgls, data = protests)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.25424 -0.32143 -0.02315  0.29313  1.63801
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## const.gls               -8.420488   0.366711 -22.962  < 2e-16 ***
## secretpol_revised.gls -0.252149   0.035191  -7.165 9.57e-13 ***
## l.ln_pop.gls             0.516274   0.022047  23.417  < 2e-16 ***
## l.ln_gdppc.gls           0.044860   0.020036   2.239 0.025230 *
## l12gr.gls               -0.004943   0.001356  -3.644 0.000272 ***
## l.lexclpop.gls          -0.042889   0.051030  -0.840 0.400712
## nbr_protest.gls          0.138695   0.017534   7.910 3.49e-15 ***
## intrastate.gls           0.191343   0.025758   7.429 1.40e-13 ***
## attempt.gls              0.110429   0.031822   3.470 0.000527 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4359 on 3245 degrees of freedom
## Multiple R-squared:  0.2234, Adjusted R-squared:  0.2212
## F-statistic: 103.7 on 9 and 3245 DF,  p-value: < 2.2e-16
```

```r
Vcl.gls <- vcovCL(gls, cluster=protests$ccode)
round(coeftest(gls, Vcl.gls), 4)
```

```
##
## t test of coefficients:
##
##                        Estimate Std. Error t value Pr(>|t|)
## const.gls               -8.4205     1.1471 -7.3409   <2e-16 ***
## secretpol_revised.gls   -0.2521     0.0844 -2.9864   0.0028 **
## l.ln_pop.gls             0.5163     0.0688  7.5047   <2e-16 ***
## l.ln_gdppc.gls           0.0449     0.0623  0.7197   0.4718
## l12gr.gls               -0.0049     0.0025 -1.9586   0.0502 .
## l.lexclpop.gls          -0.0429     0.1002 -0.4280   0.6687
## nbr_protest.gls          0.1387     0.0615  2.2561   0.0241 *
## intrastate.gls           0.1913     0.0526  3.6370   0.0003 ***
## attempt.gls              0.1104     0.0422  2.6144   0.0090 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
## The lme4 package is the more common way to go here.  It takes
## a formula of the form y~x+(1|heterogeneity).
## However, it does not work with the sandwich package, so
## we move the clubSandwich package for clustering.
## It also doesn't like the lmtest package that much
re  <- lmer(update(f1, . ~ . + (1|ccode)), data=protests)
Vcl.re <- vcovCR(re, cluster=protests$ccode, type="CR1")
coef_test(re, Vcl.re)
```

```
##              Coef. Estimate      SE t-stat d.f. (Satt) p-val (Satt) Sig.
##       (Intercept)  -8.8185 1.27158 -6.935        67.0       <0.001  ***
##  secretpol_revised  -0.2585 0.08628 -2.996        22.3       0.0066   **
##           l.ln_pop   0.5493 0.07827  7.018        60.0       <0.001  ***
##         l.ln_gdppc   0.0285 0.06664  0.427        23.1       0.6731
##              l12gr  -0.0047 0.00252 -1.870        22.8       0.0744    .
##         l.lexclpop  -0.0365 0.10105 -0.361        18.5       0.7219
##        nbr_protest   0.1307 0.06265  2.087        65.0       0.0408    *
##         intrastate   0.1898 0.05268  3.604        48.7       <0.001  ***
```

```
##             attempt   0.1109 0.04224  2.625        47.8       0.0116    *
```

```
## hausman  (with iid)
Htest <- c(within2$coefficients - re@beta[-1]) %*%
  solve(within2$cov.iid - vcov(re)[-1,-1]) %*%
  c(within2$coefficients - re@beta[-1])
pchisq(drop(Htest), df=length(within2$coefficients), lower=FALSE)
```

```
## [1] 0.0004674246
```

```
##hausman  (with clustering) but this version is sus
Htest.cl <- c(within2$coefficients - re@beta[-1]) %*%
  solve(vcov(within2) - Vcl.re[-1,-1]) %*%
  c(within2$coefficients - re@beta[-1])
pchisq(drop(Htest.cl), df=length(within2$coefficients), lower=FALSE)
```

```
## [1] 1
```

```
### Mundlak--pooled
Xnames <- colnames(pooled$model)[-1]
protests[,paste0(var.names, ".bar"):=lapply(.SD, \(x){mean(x)}),
                  by=ccode, .SDcols=var.names ]
mundlak.add <- paste(".~.",paste0("+", Xnames, ".bar",collapse = "" ))
mundlak.formula <- update(f1, mundlak.add)
mundlak <- lm(mundlak.formula, data=protests)
Vcl.m <- vcovCL(mundlak, cluster=protests$ccode)
round(coeftest(mundlak, Vcl.m), 4)
```

```
##
## t test of coefficients:
##
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            -6.1546     0.8482 -7.2564   <2e-16 ***
## secretpol_revised      -0.2716     0.0911 -2.9816   0.0029 **
## l.ln_pop                0.6411     0.1060  6.0487   <2e-16 ***
## l.ln_gdppc             -0.0180     0.0791 -0.2273   0.8202
## l12gr                  -0.0041     0.0025 -1.6233   0.1046
## l.lexclpop             -0.0128     0.1058 -0.1210   0.9037
## nbr_protest             0.1088     0.0654  1.6649   0.0960 .
```

```
## intrastate              0.1851     0.0533  3.4714   0.0005 ***
## attempt                 0.1141     0.0425  2.6828   0.0073 **
## secretpol_revised.bar   0.3140     0.1828  1.7180   0.0859 .
## l.ln_pop.bar           -0.3116     0.1130 -2.7576   0.0059 **
## l.ln_gdppc.bar          0.1318     0.0958  1.3748   0.1693
## l12gr.bar              -0.0383     0.0198 -1.9360   0.0530 .
## l.lexclpop.bar          0.0655     0.2354  0.2782   0.7809
## nbr_protest.bar         0.0648     0.0866  0.7480   0.4545
## intrastate.bar         -0.0091     0.2013 -0.0450   0.9641
## attempt.bar             1.1473     0.6343  1.8088   0.0706 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
linearHypothesis(mundlak, paste0(Xnames, ".bar=0"), vcov=Vcl.m)
```

```
## Linear hypothesis test
##
## Hypothesis:
## secretpol_revised.bar = 0
## l.ln_pop.bar = 0
## l.ln_gdppc.bar = 0
## l12gr.bar = 0
## l.lexclpop.bar = 0
## nbr_protest.bar = 0
## intrastate.bar = 0
## attempt.bar = 0
##
## Model 1: restricted model
## Model 2: Protest ~ secretpol_revised + l.ln_pop + l.ln_gdppc + l12gr +
##     l.lexclpop + nbr_protest + intrastate + attempt + secretpol_revised.bar +
##     l.ln_pop.bar + l.ln_gdppc.bar + l12gr.bar + l.lexclpop.bar +
##     nbr_protest.bar + intrastate.bar + attempt.bar
##
## Note: Coefficient covariance matrix supplied.
##
##   Res.Df Df     F  Pr(>F)
## 1   3245
```

```
## 2   3237  8 2.562 0.00876 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
### Mundlak--CRE
cre  <- lmer(update(mundlak.formula, . ~ . + (1|ccode)), data=protests)
Vcl.cre <- vcovCR(cre, cluster=protests$ccode, type="CR1")
coef_test(cre, Vcl.cre)
```

```
##                  Coef. Estimate      SE t-stat d.f. (Satt) p-val (Satt) Sig.
##             (Intercept) -6.10516 0.84238 -7.248        45.2      < 0.001  ***
##      secretpol_revised -0.27164 0.09088 -2.989        20.7      0.00707   **
##                l.ln_pop  0.64111 0.10573  6.064        41.7      < 0.001  ***
##              l.ln_gdppc -0.01798 0.07888 -0.228        18.1      0.82229
##                   l12gr -0.00409 0.00252 -1.627        22.6      0.11753
##              l.lexclpop -0.01280 0.10552 -0.121        17.5      0.90486
##             nbr_protest  0.10882 0.06520  1.669        60.3      0.10029
##              intrastate  0.18515 0.05320  3.480        48.2      0.00107   **
##                 attempt  0.11406 0.04241  2.689        47.7      0.00983   **
##  secretpol_revised.bar  0.31416 0.19237  1.633        45.4      0.10938
##            l.ln_pop.bar -0.31137 0.11380 -2.736        54.1      0.00839   **
##          l.ln_gdppc.bar  0.12956 0.09593  1.351        37.9      0.18482
##               l12gr.bar -0.02341 0.01857 -1.260        13.7      0.22858
##          l.lexclpop.bar  0.03421 0.24765  0.138        37.9      0.89086
##         nbr_protest.bar  0.07145 0.08324  0.858        46.3      0.39515
##          intrastate.bar -0.04938 0.21551 -0.229        34.1      0.82012
##             attempt.bar  1.32038 0.57089  2.313        20.6      0.03116    *
```

```
## CRE R squared
1-sum(residuals(cre)^2)/sum((protests$Protest-mean(protests$Protest))^2)
```

```
## [1] 0.7812683
```

```
##Matches the LSDV closely
max(abs(lsdv$residuals-residuals(cre)))
```

```
## [1] 0.2097183
```

```
### between
protests[, Protest.bar := mean(Protest), by=ccode]
```

```
f.btwn <- as.formula(paste("Protest.bar ~",
                           paste0(Xnames, ".bar",collapse = " + " ) ))
btwn <- lm(f.btwn, data=protests)

cbind(within2$coefficients, mundlak$coef[2:9])
```

```
##                            [,1]         [,2]
## secretpol_revised -0.271641863 -0.271641863
## l.ln_pop           0.641113791  0.641113791
## l.ln_gdppc        -0.017975958 -0.017975958
## l12gr             -0.004094911 -0.004094911
## l.lexclpop        -0.012795881 -0.012795881
## nbr_protest        0.108823234  0.108823234
## intrastate         0.185147476  0.185147476
## attempt            0.114057919  0.114057919
```

```
cbind(BtwnDiff=btwn$coef[-1]-within2$coefficients,
      mundlak$coef[10:17])
```

```
##                              BtwnDiff
## secretpol_revised.bar  0.314034175  0.314034175
## l.ln_pop.bar          -0.311639916 -0.311639916
## l.ln_gdppc.bar         0.131772527  0.131772527
## l12gr.bar             -0.038255641 -0.038255641
## l.lexclpop.bar         0.065499295  0.065499295
## nbr_protest.bar        0.064750150  0.064750150
## intrastate.bar        -0.009054607 -0.009054607
## attempt.bar            1.147308716  1.147308716
```

```
modelsummary(list("Pooled"=pooled,
                  "RE-GLS"=gls,
                  "RE-MLE"=re,
                  "LSDV"=lsdv,
                  "Within"=within2,
                  "Mundlak"=mundlak,
                  "CRE"=cre),
             vcov=list(Vcl.pooled, Vcl.gls, Vcl.re,
                       Vcl.lsdv, vcov(within2),
```

	Pooled	RE-GLS	RE-MLE	LSDV	Within	Mundlak	CRE
Secret police	−0.07	−0.25	−0.26	−0.27	−0.27	−0.27	−0.27
	(0.11)	(0.08)	(0.09)	(0.09)	(0.09)	(0.09)	(0.09)
Num.Obs.	3254	3254	3254	3254	3254	3254	3254
R2	0.416	0.223		0.783	0.782	0.447	
R2 Within					0.223		

```
                    Vcl.m, Vcl.cre),
          fmt=2,
          coef_map=c("secretpol_revised"="Secret police",
                     "secretpol_revised.gls"="Secret police",
                     "secretpol_revised.within"="Secret police"),
          gof_map=c("nobs", "r.squared", "r2.within"))
```

1.5 Two-way heterogeneity

Having considered the one-way heterogeneity model to some extent, we may want start considering extensions. Given that panels are often seen as N separate time-series we may want to start by thinking about a model of the form

$$y_{it} = \beta' x_{it} + \alpha_i + f(t) + \varepsilon_{it}.$$

As in the above format we can consider the heterogenity as functions of observable and unobservable factors such that

$$\alpha_i = \alpha + \gamma' z_i + u_i$$

There are many ways to think about time here.
The easiest is a simple time trend: $f(t) = \tau t$. This could be made more flexible by using polynomials $f(t) = \tau_1 t + \tau_2 t^2 + \ldots$ or splines. Alternatively, we could do a time trend by individual $f_i(t) = \tau_i t$, in which case we would interact t with the unit-dummies.

These kind of functional forms can be appealing, but they tend to require a relatively strong assumption on how time works in the empirical model. Should it be linear? Some kind of cycle? Cycles may get weird. As such a non-parametric form based on the full specification above may be preferred, as in

$$f(t) = \tau + \psi' w_t + v_t.$$

This approach is called two-way heterogeneity or the two-way fixed effects model. In matrix form we can write this for a balanced panel as

$$y = \begin{bmatrix} X & (I_N \otimes 1_T) & (1_T \otimes I_N) \end{bmatrix} \theta + \varepsilon,$$

where $\theta = (\beta, \alpha_i, \tau_t)$. For examples on Kronecker products

```
N <- 3
T <- 2
diag(N) %x% rep(1, T)
```

```
##      [,1] [,2] [,3]
## [1,]    1    0    0
## [2,]    1    0    0
## [3,]    0    1    0
## [4,]    0    1    0
## [5,]    0    0    1
## [6,]    0    0    1
```

```
rep(1, T) %x% diag(N)
```

```
##      [,1] [,2] [,3]
## [1,]    1    0    0
## [2,]    0    1    0
## [3,]    0    0    1
## [4,]    1    0    0
## [5,]    0    1    0
## [6,]    0    0    1
```

Now when we believe that two-way heterogeneity is present we cannot ignore either form. Omitted variables are of course one problem, but even with no correlation we have a problem if either dimension is small. For example, consider a large N survey over a small number of

waves T and consider the one-way within estimator:

$$\begin{aligned}
\hat{\beta}_w &= \left[\sum_i \sum_t (x_{it} - \bar{x}_i)(x_{it} - \bar{x}_i)'\right]^{-1} \left[\sum_i \sum_t (x_{it} - \bar{x}_i)(y_{it} - \bar{y}_i)\right] \\
&= \beta + \left[\frac{1}{NT}\sum_i \sum_t (x_{it} - \bar{x}_i)(x_{it} - \bar{x}_i)'\right]^{-1} \left[\frac{1}{NT}\left(\sum_i \sum_t (x_{it} - \bar{x}_i)(\tau_t - \bar{\tau})\right.\right. \\
&\qquad \left.\left. + \sum_i \sum_t (x_{it} - \bar{x}_i)(\varepsilon_{it} - \bar{\varepsilon}_i)\right)\right] \\
&= \beta + \left[\frac{1}{NT}\sum_i \sum_t (x_{it} - \bar{x}_i)(x_{it} - \bar{x}_i)'\right]^{-1} \left[\frac{1}{NT}\left(\sum_i \sum_t (x_{it} - \bar{x}_i)(\tau_t - \bar{\tau}) + 0\right)\right]
\end{aligned}$$

So far so good, but with a little algebra we get to

$$\frac{1}{NT}\sum_i \sum_t (x_{it} - \bar{x}_i)(\tau_t - \bar{\tau}) = \frac{1}{T}\sum_t (\bar{x}_t - \bar{x})(\tau_t - \bar{\tau}).$$

This will converge to its expected value (zero if x is uncorrelated with the time effects), but this convergence is in T! If T is relatively small, then we can't rely on that. To put this another way, even if they are uncorrelated with the observables, time effects can still bias the estimates of β if T is not large!

The consequence of this is that unless you believe that the time effects are constant $\tau_t = \bar{\tau}$ for all t, if this dimension is small, then we should consider time heterogeneity as an important bias to control for.

We can do this in the same three ways we described above:

1. A different within transformation:

$$D_2 = \underbrace{I_{NT} - (I_N \otimes 1_T 1_T'/T)}_{\text{Unit demeaning}} - \underbrace{(1_N 1_N'/N \otimes I_T)}_{\text{Time means}} + \underbrace{\frac{1}{NT} 1_{NT} 1_{NT}'}_{\text{overall mean}}.$$

   Note that here (the balanced case) we have the original group-wise demeaning, then time-wise demeaning and then we add back in the overall mean, as in

$$D_2 X = \left[x_{it} - \bar{x}_i - \bar{x}_t + \bar{x}\right].$$

   This transformation is more involved with unbalanced panels.
2. Dummies: As before, just include dummies for each i and each t. 1 of these will need to be removed to avoid colinearity

3. CRE: When the panel is balanced, then CRE with time means and group means will still be equivalent. This equivalence does not hold for unbalanced panels.

The within transformation in unbalanced panels is slightly convoluted, but we can see how it maps into the above.

$$D_2' = M - M\Delta_T[\Delta_T' M \Delta_T]\Delta_T' M$$
$$M = I_{NT} - \Delta_N[\Delta_N'\Delta_N]^{-1}\Delta_N'.$$

Here, Δ_N and Δ_T are matrices of unit and time dummies, respectively. We remove the first (or any) column from Δ_T to avoid colinearity. Note that M here is the unit-demeaning matrix for the whole sample (diagonal binding the M_is).

When the panel is balanced,

$$\Delta_N[\Delta_N'\Delta_N]^{-1}\Delta_N' = (I_N \otimes 1_T 1_T'/T),$$

which gives us a block diagonal matrix of $1/T_i$, and

$$M\Delta_T[\Delta_T' M \Delta_T]\Delta_T' M = (1_N 1_N'/N \otimes I_T) + \frac{1}{NT} 1_{NT} 1_{NT}'.$$

In the unbalanced case we get weighted averages for the time and overall means based on how often they appear in the sample.

1.5.1 Asymptotics in T

While we're considering the different dimensions of the panel, we should also be clear about fixed-N asymptotics. In survey data and many other contexts, fixed-T-large-N makes sense. However, in other parts of political science we often have a fixed (or fairly fixed) N. For example, the number of U.S. states or countries of the world don't increase all that often and are fairly static in many cases for which we collect data.

In these cases, it may make more sense to think about large T asymptotics. After all, in country-year data we typically have a fairly fixed N, but T is increasing as we move forward in time and data collection continues. So what does it mean to think about the panel estimators in that context?

The pooled estimator should now be rewritten as

$$\hat{\theta}_p = \theta + \left[\frac{1}{T}\sum_{t=1}^{T}\left(\frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_{it}\mathbf{x}_{it}'\right)\right]^{-1}\left(\frac{1}{T}\sum_{t=1}^{T}\left(\frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_{it}\varepsilon_{it}\right)\right).$$

Before we can do anything with this, we'll need to add a few assumptions. First, instead of allowing for arbitrary correlation within-units we'll impose some constraints on the time series

Assumption A6 *The sequence* $(\mathbf{x}_t, \varepsilon_t)$ *is strictly stationary and ergodic*

Note that here $\mathbf{x}_t = (\mathbf{x}_{1t}, \mathbf{x}_{2t}, \ldots, \mathbf{x}_{Nt})$ and likewise for ε_t.

Assumption A7 *The matrix* $\mathrm{E}[\frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_{it}\mathbf{x}_{it}']$ *has full rank.*

To review, a sequence y_t is (strictly) *stationary* if the joint distribution of $(y_t, \ldots y_{t+k})$ is independent of both t and k. This basically means that distribution of y_t does not change with time so its mean and variance are constant, but also that the relationships between parts of the time series are constant over time. For example, the covariance between y_1 and y_3 is the same as y_5 and y_7.

Some example of stationary series include $y_t = x_t + \theta x_{t-1}$, $y_t = x_t$, and $y_t = x$, where x_t is iid with $|\theta| < 1$, $\mathrm{E}[x_t] = 0$ and x is a single realization of a random variable.

A stationary series y_t is **erogodic** if and only if, well it's complicated and takes awhile to really explain. However, t its core it means that if there are any events that are invariant to t (i.e., is the maximum of y_t positive?) then the probability of these events must by 0 or 1. In the case of $y_t = Z$ we have y_t is a single draw from a random variable. If Z is standard normal, then the invariant event is the maximum of y_t positive? has probability 0.5. As such this sequence is not ergodic. At its core, an ergodic sequence can never get "stuck." An ergodic y_t will eventually visit every value in its support if the sequence lasts long enough and it can move from any part of its support to any other with positive probability.

To make life easier, we will also assume that y_t is *mixing*.
This means that as ℓ increases the $\mathrm{Cov}(y_t, y_{t-\ell})$ goes to 0. As the time between points increases, they provide less and less information about each other. Note that mixing implies ergodicity, but not vice-versa.

Of import to us is the following theorem

Theorem 7 *Let* y_t *be a strictly stationary and ergodic random variable and let* f *be a continuous function. Then* $X_t = f(y_t, y_{t-1}, \ldots)$ *is also strictly stationary and ergodic.*

This theorem tells us that stationarity is preserved by continuous transformations that consider some or part of the history of y_t. Don't lose any sleep over it other than to remember the intuition part.

What this gives us now is a time-series version of a law of large numbers

Theorem 8 *(Ergodic LLN) Let y_t be stationary and ergodic with $\mathrm{E}[y_t] < \infty$, then $\frac{1}{T}\sum_{t=1}^{T} y_t \xrightarrow{p} \mathrm{E}[y_t]$.*

This gives us something to work with now

$$\frac{1}{T}\sum_{t=1}^{T}\left(\frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_{it}\mathbf{x}_{it}'\right) \xrightarrow{p} \mathrm{E}\left[\frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_{it}\mathbf{x}_{it}'\right]$$
$$\frac{1}{T}\sum_{t=1}^{T}\left(\frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_{it}\varepsilon_{it}\right) \xrightarrow{p} \mathrm{E}\left[\frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_{it}\varepsilon_{it}\right]$$
$$\mathrm{E}\left[\frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_{it}\varepsilon_{it}\right] = \mathrm{E}_{\mathbf{x}_{it}}\left[\frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_{it}\,\mathrm{E}\left[\varepsilon_{it}|\mathbf{x}_{it}\right]\right] = 0$$

From here, the usual applications of Slutsky's theorem follows and we get that pooled OLS is consistent in T under Assumptions A1.A, A2, A3′, A6, & A7 the pooled estimator is consistent and surely exists for large enough T, where we replace A3 with

Assumption A3′ $\mathrm{E}[\varepsilon_{it}|\mathbf{x}_{it}] = 0$.

If we want to include strict-within unit exogeneity (perlaps unlikely as T increases), then we also get unbiased estimates.

Assumption A8 *Additional technical assumptions that allow us to use a central limit theorem for dependent data*

We now present a central limit theorem for stationary mixing sequences

Theorem 9 *Let z_t be strictly stationary and mixing with $\mathrm{E}[z_t] = 0$ and some other conditions. Then*

$$\sqrt{T}\left(\frac{1}{T}\sum_{t=1}^{T} z_t\right) \xrightarrow{d} N(0, \Sigma_T)$$

For ease let $z_t = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_{it}\varepsilon_{it}$ be stationary and mixing. This gives us the following to work with

$$\sqrt{T}\frac{1}{T}\sum_{t=1}^{T}\frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_{it}\varepsilon_{it} = \sqrt{T}\frac{1}{T}\sum_{t=1}^{T} z_t \xrightarrow{d} N(0, \Sigma_T)$$
$$\Sigma_T = \mathrm{Var}\left(T^{-1/2}\sum_{t=1}^{T} z_t\right)$$

Using some time series results for stationary and ergodic series we can write this as

$$\Sigma_T = \lambda(0) + \sum_{\ell=1}^{T} \left(1 - \frac{\ell}{T}\right) (\lambda(\ell) + \lambda(\ell)'),$$

where $\lambda(\ell)$ are the covariances matrix of the tth observation with the ℓth lag

$$\lambda(\ell) = \sum_{t=\ell+1}^{T} \sum_{i=1}^{N} \mathbf{x}_{it} \varepsilon_{it} \varepsilon_{it-\ell} \mathbf{x}'_{it-\ell},$$

This make $\lambda(0)$ the contemporary variance, which in this case is the meat of a cluster-robust covariance matrix where we cluster on time.

As $T \to \infty$, full consideration of Σ_T becomes unbearable and will contain many irrelevant lags that add little-to-no information and probably some excess noise because there aren't as many lags of that length to average over.

To avoid this we can exploit the diminishing nature of the dependency (i.e., the mixing component) to get

$$\widehat{\Sigma}_T(L) = \hat{\lambda}(0) + \sum_{\ell=1}^{L} \left(1 - \frac{\ell}{L+1}\right) \left(\hat{\lambda}(\ell) + \hat{\lambda}(\ell)'\right).$$

However, we now have to choose a maximum lag value have to choose L and we should choose L such that it increases with T, for example $T^(1/4)$ is a frequent default and not a bad starting point, other more thoughtful options exist.

The whole covariance estimator is then

$$\widehat{\text{avar}}(\hat{\theta}; L) = [\mathbf{X}'\mathbf{X}]^{-1} \widehat{\Sigma}_T(L) [\mathbf{X}'\mathbf{X}]^{-1},$$

note that when $L = 0$, this simplifies into a covariance matrix that is clustered by time. Similar analysis will demonstrate this for the within and RE estimators. This particular variance matrix is sometimes called the Driscol-Kraay covariance matrix after their 1998 article. Note that because the baseline matrix clusters on time, that it allows for arbitrary correlation cross-sectionally (partially relaxing the assumption of iid units), while making the most of the long-T time series within each unit.

Note that if you have large N and believe that the N units are iid, then you're probably better off with the clustered variance matrix above as it allows for arbitrary within-unit correlations, but this gives you something to do in the case where T is large and N is not.