Panel Data

Today's Outline

- Panel Data models
 - Organization
 - First differences
 - Fixed effects
 - Random effects
 - Hausman test

Assignment due tonight (November 30th!)

A panel Data Model

- Panel data can be boiled down to data that takes multiple observations for the same individuals over different time periods
- We can index every observation by the individual who was observed (i) and the time the observation was recorded (t)
- A generic panel model takes the following form:

$$y_{it} = eta_0 + eta_1 x_{1it} + eta_2 x_{2it} + \ldots + eta_k x_{kit} + a_i + u_{it}$$

Panel data Example

- Suppose we have three individuals (Larry, Dean, Sarah) observed once a year from 2000 to 2009
- This is an example of a panel data set

```
# We have three individuals
i = ["Larry", "Dean", "Sarah"]

# We take observations once a year
t = np.arange(2000, 2010)

# each individual is observed once a year
panel_example = pd.DataFrame(itertools.product(t,i), columns = ["year", "person"])

panel_example.head()
```

 year
 person

 0
 2000
 Larry

 1
 2000
 Dean

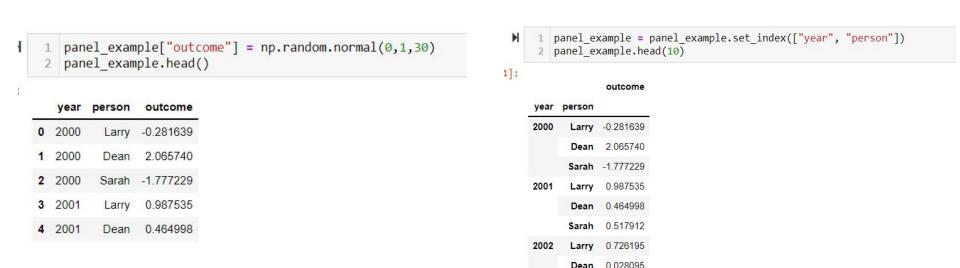
 2
 2000
 Sarah

 3
 2001
 Larry

 4
 2001
 Dean

Organizing Panel Data

- Some synthetic Y values are recorded for each observation
- To work with most panel modelling functions, we need to set the index of our panel dataset as a dual index with the (i) and (t) variables
- The df.set_index() function accepts a list of columns as an arguments, where the columns are the indices

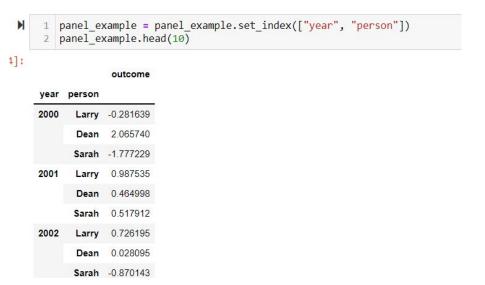


Sarah -0.870143

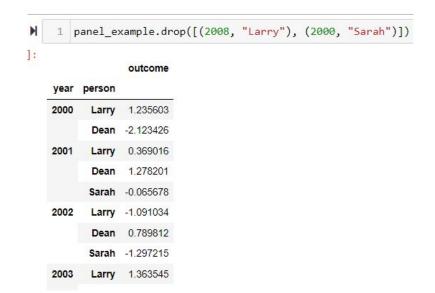
Balanced Panel

- Since we have an observation for each individual for all periods we call this a "balanced panel"
- If for any reason this wasn't the case then the panel would be unbalanced

Balanced



Unbalanced



A panel Data Model

A generic panel model takes the following form:

$$y_{it} = eta_0 + eta_1 x_{1it} + eta_2 x_{2it} + \ldots + eta_k x_{kit} + a_i + u_{it}$$

 Note that we have broken out our usual error term into a component that is dependent on time and one that is not

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

Unfortunately endogeneity is often a problem in panel data.
 Specifically, when the part of the errors for an individual that don't change across time are correlated with X and Y

Panel Data Models

- Endogeneity is a problem we have encountered before
- One form of endogeneity will occur when our regressors (x) are correlated with the unobserved error term (a_i)
- For example, suppose we are studying the effect of union membership on wages
 - Question: Do workers earn more as a result of union membership or are there factors that would cause them to earn more anyways? (more skilled, experienced, etc).
- With proper techniques (fixed effects), panel data allows us to control for individual confounders that don't change (or change very slowly) over time:
 - o Sex, IQ, ethnicity, location, etc.
- We can also adjust our standard errors for serial correlation that naturally occurs when we observe the same people over time (using random effects)

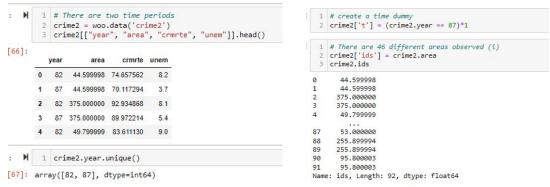
$$log(wage)_{it} = \beta_0 + \beta_1 Union_{it} + a_i + u_{it}$$

First Differences

Suppose we are trying to answer the question of whether unemployment causes an increase in the crime rate

$$crmrte_{it} = \beta_0 + \beta_1 unem_{it} + ai + u_{it}$$

We have a panel data set that observes different areas over two time periods



If we only have two time periods, we can take advantage of panel data simply by taking the difference between our
variables between the first and second observation for each individual

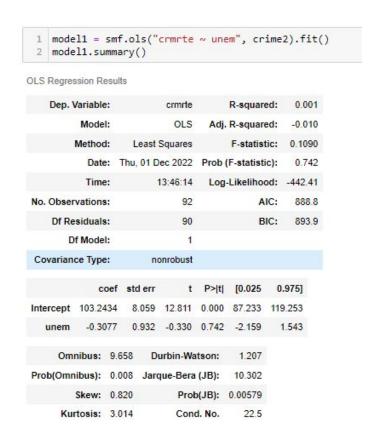
$$crmrte_{i2} - crmrte_{i1} = (eta_0 - eta_0) + eta_1(unem_{i2} - unem_{i1}) + (ai - ai) + u_{i2} - u_{i1})$$

This will eliminate any confounding variables that don't change over time

$$\Delta crmrte_{it} = \beta_1 \Delta unem_{it} + \Delta u_{it}$$

First Differences vs OLS

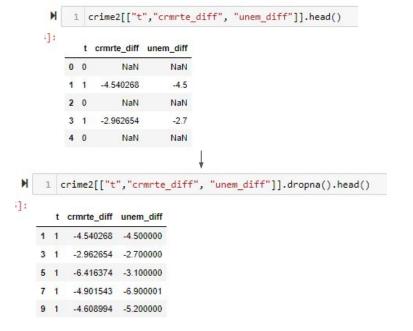
We can see very large changes between the pooled estimator and the first differences estimator



```
fdiff = smf.ols("crmrte_diff ~ unem_diff", crime2).fit()
  2 fdiff.summary()
OLS Regression Results
    Dep. Variable:
                         crmrte diff
                                         R-squared:
                                                       0.127
           Model:
                              OLS
                                     Adj. R-squared:
                                                       0.107
          Method:
                     Least Squares
                                          F-statistic:
                                                       6.384
                                   Prob (F-statistic):
            Date: Thu. 01 Dec 2022
                                                     0.0152
            Time:
                          13:46:14
                                     Log-Likelihood: -202.17
 No. Observations:
                                46
                                                AIC:
                                                       408.3
                                44
                                                BIC:
                                                       412.0
     Df Residuals:
        Df Model:
 Covariance Type:
                          nonrobust
                                         [0.025 0.975]
              coef
                    std err
                                   P>|t|
                      4.702 3.276 0.002
  Intercept 15,4022
                                          5.926 24.879
 unem_diff 2.2180
                     0.878 2.527 0.015
                                         0.449
                                                3.987
      Omnibus: 2.636
                         Durbin-Watson:
 Prob(Omnibus): 0.268 Jarque-Bera (JB): 2.255
          Skew: 0.539
                               Prob(JB): 0.324
       Kurtosis: 2.883
                              Cond. No.
                                          8.70
```

First Differences in linearmodels

- The linearmodels package will contain all of our panel estimators
- Each of these linearmodels functions require that our panel data is indexed correctly before estimation
- Note that "t" here functions as an intercept after differencing and represents the time trend



```
1 import linearmodels as plm
  1 crime2 = crime2.set index(['ids', 'year'])
 1 plm.FirstDifferenceOLS.from_formula(formula = 'crmrte ~ t + unem', data = crime2).fit()
C:\Users\kunzn\anaconda3\lib\site-packages\linearmodels\shared\utility.py:187: FutureWarning:
all arguments of MultiIndex.set levels except for the argument 'levels' will be keyword-only
  df.index = df.index.set_levels(final_levels, [0, 1])
FirstDifferenceOLS Estimation Summary
    Dep. Variable:
                            crmrte
                                            R-squared: 0.1961
       Estimator: FirstDifferenceOLS
                                  R-squared (Between):
No. Observations:
                                     R-squared (Within): 0.1961
                  Thu. Dec 01 2022
                                    R-squared (Overall):
                                                      0.4041
                          13:50:35
                                         Log-likelihood -202.17
           Time:
   Cov. Estimator:
                        Unadjusted
                                                       5.3653
                                            F-statistic:
                               46
                                                      0.0082
         Entities:
                                               P-value
        Avg Obs:
                           2.0000
                                           Distribution: F(2.44)
        Min Obs:
                           2.0000
        Max Obs:
                           2.0000
                                     F-statistic (robust):
                                                       5.3653
                                               P-value 0.0082
                               2
                                          Distribution: F(2,44)
    Time periods:
                            46.000
        Avg Obs:
        Min Obs:
                            46.000
        Max Obs:
                           46.000
Parameter Estimates
                           T-stat P-value Lower CI
                                                     24.879
```

3.9872

unem

2.2180

0.8779 2.5266 0.0152

Fixed Effects

- Fixed effects is another, similar, method of controlling for the unobserved time-invariant effects
- For two periods, the estimates obtained by fixed effects are the same as the first differences estimator
- The fixed effects model can be estimated equivalently in two ways, first by demeaning each observation by for the individuals:

$$y_{it} = \beta_0 + \beta_1 x_{it1} + \dots + \beta_k x_{itk} + a_i + u_{it}; \qquad t = 1, \dots, T; \qquad i = 1, \dots, n,$$

$$\bar{y}_i = \beta_0 + \beta_1 \bar{x}_{i1} + \dots + \beta_k \bar{x}_{ik} + a_i + \bar{u}_i$$

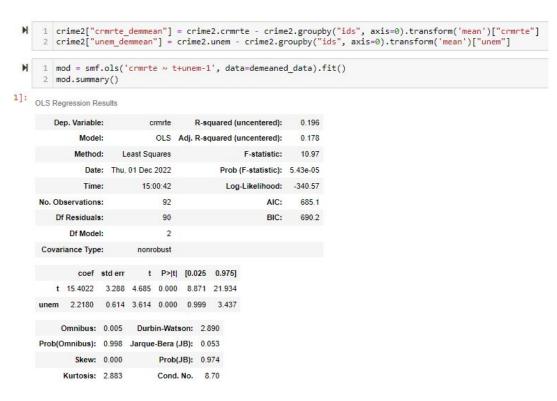
$$\ddot{y}_{it} = y_{it} - \bar{y}_i = \beta_1 \ddot{x}_{it1} + \dots + \beta_k \ddot{x}_{itk} + \ddot{u}_{it},$$

- Note that doing this manually will lead to incorrect standard errors
- We can also just include a dummy in our regression for each individual (this is computationally less efficient, and creates an uglier output, but will give the same result)

$smf.ols('Y \sim x1 + C(person)-1', data = data).fit().summary()$

Within Estimator

- We can also use differencing across many time periods
- The within estimator subtracts the mean of our variables from each period
- The smf.ols() function adds an intercept automatically that we can remove with "-1"



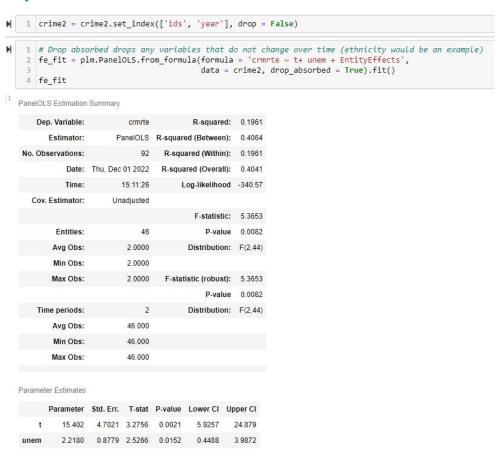
$$y_{it} = \beta_0 + \beta_1 x_{it1} + \dots + \beta_k x_{itk} + a_i + u_{it}; \qquad t = 1, \dots, T; \qquad i = 1, \dots, n,$$

$$\bar{y}_i = \beta_0 + \beta_1 \bar{x}_{i1} + \dots + \beta_k \bar{x}_{ik} + a_i + \bar{u}_i$$

$$\ddot{y}_{it} = y_{it} - \bar{y}_i = \beta_1 \ddot{x}_{it1} + \dots + \beta_k \ddot{x}_{itk} + \ddot{u}_{it},$$

Within Estimator (linearmodels)

- The PanelOLS.from_formula() function automatically can implement fixed effects
- Add EntityEffects to the formula
- PanelOLS does not automatically add an intercept
- drop_absorbed will remove variables from the regression that do not change over time



Dummy Variable Regression

- Fixed effects can be estimated equivalently by including a dummy variable for each (less one) (i) in the sample
- We can test if fixed effects are necessary by running an F-test on the individual dummy variables

1 5	smf.ols(for	rmula =	'crmrte	~ t+ u	nem +	C(ids)',	data	= crime2)).fit().summary(
OLS R	egression Res	ults							
D	ep. Variable:		crmrte	R-s	squared:	0.891			
	Model:		OLS	Adj. R-	squared:	0.774			
	Method:	Least	Squares	F-	statistic:	7.642			
	Date:	Thu, 01 D	ec 2022 F	rob (F-s	tatistic):	1.70e-10	ũ		
	Time:		11:44:37	Log-Lik	elihood:	-340.57			
No. O	bservations:		92		AIC:	777.1			
C	Of Residuals:		44		BIC:	898.2			
	Df Model:		47						
Cova	ariance Type:	no	onrobust						
			coe	f std er	r	t P> t	[0.025	0.975]	
		Intercept		15.51			4.473	67.010	
C(ids)[T.17.7999992	(57)35000050				4 0.000	80.489	140.917	
CAROLINA CO)[T.18.8999996					3 0.002	18.444	77.073	
)[T.20.7999992						57.768	119.175	
70.00)[T.21.8999996				9 0.925	0.360	-16.107	43.461	
)[T.24.1000003			14.23		0.001	23.105	80.467	
C(ids)[T.24.2000007	76 2 939453]	24.2242	15.000	6 1.614	0.114	-6.019	54.467	
C(ids)[T.25.2999992	237060547]	48.2313	14.97	8 3.220	0.002	18.046	78.417	
C(ids)[T.27.3999996	18530277]	30.3498	14.79	1 2.052	0.046	0.540	60.159	
C(id	s)[T.34.200000	76293945]	43.7282	14.54	5 3.006	0.004	14.414	73.042	
	C(i	ds)[T.604.0	52.670	7 15.050	0 3.500	0.001	22.340	83.002	
			t 15.4022				5.926	24.879	
		unen	1 2.2180	0.87	8 2.527	0.015	0.449	3.987	
	Omnibus:	0.005	Ourbin-Wat	son: 3.	413				
Prot	Omnibus):	0.998 Ja	rque-Bera (JB): 0.0	053				
	Skew:	-0.000	Prob	JB): 0.9	974				
	Kurtosis:	2.883	Cond	. No. 4	125.				

Random Effects

Random Effects is another panel data method that is only appropriate when:

$$Cov(x_{iti}, a_i) = 0, \quad t = 1, 2, ..., T; j = 1, 2, ..., k.$$

- That means we have no endogeneity!
- However the errors in our model will be serially correlated across time, since even if
 u_it is uncorrelated with itself and a_i, the covariance of a_i with itself is non-zero
- More precisely:

$$Cov(a_i + u_{it}, a_i + u_{is}) = Var(a_i)^2$$

 This violates our basic assumption for OLS estimators for that the error is not correlated with itself

Random Effects Original Regression

- Here we fit the model again using OLS to compare with the random effects model
- Note that we can include an intercept since we do not need to eliminate time invariant parts of the model
- We can include a time dummy to account for changes between each period as well
- Note that neither are statistically significant
 - Direction of unem makes sense with t included
 - Standard errors are large

```
mod = smf.ols('crmrte ~ t+ unem-1+1', data=crime2).fit()
mod.summary()
OLS Regression Results
    Dep. Variable:
                                                        0.012
                             crmrte
                                          R-squared:
                               OLS
                                                        -0.010
           Model:
                                      Adj. R-squared:
                      Least Squares
                                                       0.5501
         Method:
                                           F-statistic:
                  Wed, 06 Dec 2023
                                                        0.579
                                    Prob (F-statistic):
                           16:51:17
                                      Log-Likelihood:
                                                       -441.90
            Time:
                                92
                                                        889.8
No. Observations:
                                                AIC:
                                89
                                                        897.4
     Df Residuals:
                                                 BIC:
        Df Model:
                                  2
 Covariance Type:
                          nonrobust
                                                  0.9751
                   std err
                                   P>ltl
                                         [0.025
                    12.739 7.333
                                  0.000
                                                118,733
           7.9404
                     7.975 0.996
                                 0.322
                                         -7.906
                                                  23.787
                                                   2.788
            0.4265
                     1.188
                           0.359
                                  0.720
                                         -1.935
      Omnibus: 8.350
                         Durbin-Watson:
                                           1.157
 Prob(Omnibus): 0.015
                       Jarque-Bera (JB):
                                          8.771
         Skew: 0.756
                               Prob(JB): 0.0125
                 2.935
                               Cond. No.
                                            40.1
```

Random Effects

- Random effects estimates the structure of our variance using a type of FGLS
- This procedure corrects our standard errors
- As long as our assumption about the covariance between a and x being 0 holds, then Random Effects will produce better results than OLS or Fixed Effects
 - This can be an admittedly strong assumption
- Random Effects may also be estimated in using plm



Dep. Variable:	crmrte	R-squared:	0.0927
Estimator:	RandomEffects	R-squared (Between):	-0.0320
No. Observations:	92	R-squared (Within):	0.1911
Date:	Thu, Dec 01 2022	R-squared (Overall):	-0.0017
Time:	11:56:44	Log-likelihood	-372.87
Cov. Estimator:	Unadjusted		
		F-statistic:	4.5472
Entities:	46	P-value	0.0132
Avg Obs:	2.0000	Distribution:	F(2,89)
Min Obs:	2.0000		
Max Obs:	2.0000	F-statistic (robust):	4.5472
		P-value	0.0132
Time periods:	2	Distribution:	F(2,89)
Avg Obs:	46.000		
Min Obs:	46.000		
Max Obs:	46.000		

Parameter Estimates

	Parameter	Std. Err.	T-stat	P-value	Lower CI	Upper CI
Intercept	80.031	9.2597	8.6429	0.0000	61.632	98.430
t	13.487	4.4767	3.0128	0.0034	4.5922	22.382
unem	1.7583	0.8077	2.1767	0.0321	0.1533	3.3632

id: 0x198c37e36a0

Hausman Test

- The Hausman Test helps us determine whether we have a problem with endogeneity
- The null hypothesis is that the individual effects are exogenous
- If we reject the null then individual effects are endogenous and we should use Fixed effects
- The Hausman simply tests whether there is a statistically significant difference between the coefficients estimated by RE and FE
- Including important omitted variables will change our estimated coefficients, so if the individual effects are important then the coefficients on RE and FE should be very different

```
def PnlHausman(fe_fit, re_fit):
    # pull out the variances and parameters for test
    Dcov = fe_fit.cov - re_fit.cov.iloc[1:, 1:]
    dparams = fe_fit.params - re_fit.params[1:]
    # get the test statistic
    Chi2 = dparams.dot(np.linalg.inv(Dcov)).dot(dparams)
    # calculate the degrees of freedom
    dof = re_fit.params.size - 1
    # calculate the p-value
    pvalue = stats.chi2(dof).sf(Chi2)
    return(Chi2, dof, pvalue)
```

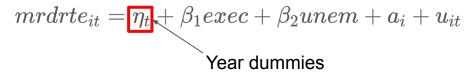
```
# Takes in a fitted fixed effects model and a fitted random effects model
# The null hypthesis is that the individual effects are exogenous
PnlHausman(fe_fit, reg_re)
```

(0.08940229523166991, 1, 0.7649383929396042)

Simulated Data Demonstration

Panel Exercise

- Download the MURDER dataset from wooldridge
- Estimate the following pooled ols model



- Estimate the same model again and include fixed effects then random effects
- (Optional) Run the hausman test for endogeneity