

MGMTMFE 431:

Data Analytics and Machine Learning

Topic 2: Panel regressions

Spring 2025

Professor Lars A. Lochstoer

From Topic 1: Visualization

1. Reduce noise

- E.g., portfolio sorts, as long as possible samples. Realized returns very noisy, never looks good to plot realized versus predicted at the stock level and over short samples

2. Reduce dimensionality

- Plots should typically be 2-D. 3-D almost never looks good
- Can use multiple lines, multiple panels (facets) to show three dimensional relations
- Perhaps add different colors, line-types for fourth and fifth dimensions -- this is pushing it.

As data scientists, our job to (1) understand which (new) dimensions are important, (2) come up with an economic rationale for why something works (a “story”), and (3) to show clearly and convincingly (with plots) what is going on in the data.

- Part of “*convincing*” is to show plots of *implementable strategies* likely to *survive transaction costs*
 - E.g., *value-weighting within portfolio*, calculate approx. *transaction costs*, *sorts based on information available at time t* when considering returns from t to $t+1$

From Topic 1: Visualization

Code from last time showed how to:

1. Create quantile sorts (e.g., decile or vingtile) at each t
2. Value-weight returns within each portfolio each period from t to $t+1$
3. Then take (equal-weighted) average across all years (weight time equally)

Also, showed how to do conditional sorts

- E.g., sort conditional on a size quantile (in this case, just small and large)

You can copy this procedure for ***your signal***

- Interact with size, industry, etc.
- Control for other, known trading strategies
- Today, we will do the latter – i.e., assess ***marginal significance, value added***

Topic 2: Panel regressions

- a. Marginal value: Multiple Regression vs. Simple Regression
- b. Omitted Variables and Fama-MacBeth
- c. Panel regression overview
- d. Panel regressions in detail
 - 1) Predicting firm earnings
 - 2) Clustered standard errors
 - 3) Fixed Effects
 - 4) Predicting firm-level return variance

a. MR (multiple regression) vs. SR (simple regression)

Here, we will consider multiple regressions and the well-known *Omitted Variable Bias*

The overall concept: How to establish that your proposed trading signal (e.g., a sentiment index from social media sites) is **valuable above and beyond** other well-known trading signals

- Value in economics is about **marginal contribution**

First, let's run some simple Fama-MacBeth regressions

- Returns regressed only on InBM first, then add other signals

a. Fama-MacBeth using lnBM (value)

Hard-code Fama-MacBeth procedure:

```
# define function that returns OLS coefficients from fitted regression
def ols_coef(x, formula): return sm.ols(formula,data=x).fit().params

# the below runs regression ExRet ~ lnBM by year, including intercept
res = (StockRetAcct_DF.groupby('year').apply(ols_coef,'ExRet ~ lnBM'))

print('Mean Return: ', str(res['lnBM'].mean())+'\n',
      'Std Dev: ', str(res['lnBM'].std())+'\n',
      'Sharpe Ratio ', str(res['lnBM'].mean()/
                            res['lnBM'].std())+'\n',
      't-stat: ', str(35**.5*(res['lnBM'].mean()/
                                res['lnBM'].std())), sep="\n")
```

Mean Return:
0.0147

Std Dev:
0.0916

Sharpe Ratio
0.1600

t-stat:
0.9465

B/M sorted portfolio not statistically significant in this sample!

- Mean return not the same as in portfolio sort as not the same portfolio and different scale (leverage)

Sharpe ratio is low

- not affected by leverage

a. Fama-MacBeth and Portfolio Weights

- Here, I derive the portfolio weight expression to make the connection between Fama-MacBeth regressions and portfolio sorts 100% clear.
- Consider the simple cross-sectional for a particular time t :

$$R_{i,t} = \lambda_{0,t} + \lambda_{1,t}x_{i,t-1} + \varepsilon_{i,t}$$

- Define:

$$\underbrace{X_{t-1}}_{N \times 2} = \begin{bmatrix} 1 & x_{1,t-1} \\ \vdots & \vdots \\ 1 & x_{N,t-1} \end{bmatrix}, \quad \underbrace{R_t}_{N \times 1} = \begin{bmatrix} R_{1,t} \\ \vdots \\ R_{N,t} \end{bmatrix}, \quad \text{then} \quad \begin{bmatrix} \lambda_{0,t} \\ \lambda_{1,t} \end{bmatrix} = (X_{t-1}' X_{t-1})^{-1} X_{t-1}' R_t$$

a. Fama-MacBeth and Portfolio Weights

- Let's look inside the expression: $(X_{t-1}' X_{t-1})^{-1} X_{t-1}' R_t$
- First:

$$X_{t-1}' X_{t-1} = N \begin{bmatrix} 1 & \frac{1}{N} \sum_{i=1}^N x_{i,t-1} \\ \frac{1}{N} \sum_{i=1}^N x_{i,t-1} & \frac{1}{N} \sum_{i=1}^N x_{i,t-1}^2 \end{bmatrix}, \quad (\text{check this yourself!})$$

Then (again, check this yourself):

$$(X_{t-1}' X_{t-1})^{-1} = \frac{1}{\frac{1}{N} \sum_{i=1}^N x_{i,t-1}^2 - \left(\frac{1}{N} \sum_{i=1}^N x_{i,t-1} \right)^2} \begin{bmatrix} \frac{1}{N} \sum_{i=1}^N x_{i,t-1}^2 & -\frac{1}{N} \sum_{i=1}^N x_{i,t-1} \\ -\frac{1}{N} \sum_{i=1}^N x_{i,t-1} & 1 \end{bmatrix}$$

a. Fama-MacBeth and Portfolio Weights

- Define:

$$E_N[x_{i,t-1}] = \frac{1}{N} \sum_{i=1}^N x_{i,t-1}, \quad Var_N[x_{i,t-1}] = \frac{1}{N} \sum_{i=1}^N x_{i,t-1}^2 - \left(\frac{1}{N} \sum_{i=1}^N x_{i,t-1} \right)^2$$

- Then, we can write:

$$(X_{t-1}' X_{t-1})^{-1} = \frac{1}{N Var_N[x_{i,t-1}]} \begin{bmatrix} E_N[x_{i,t-1}^2] & -E_N[x_{i,t-1}] \\ -E_N[x_{i,t-1}] & 1 \end{bmatrix}$$

- Then:

$$(X_{t-1}' X_{t-1})^{-1} X_{t-1}' = \frac{1}{N Var_N[x_{i,t-1}]} \begin{bmatrix} E_N[x_{i,t-1}^2] & -E_N[x_{i,t-1}] \\ -E_N[x_{i,t-1}] & 1 \end{bmatrix} \begin{bmatrix} 1 & \cdots & 1 \\ x_{1,t-1} & \cdots & x_{N,t-1} \end{bmatrix}$$

a. Fama-MacBeth and Portfolio Weights

- Let's focus on the second row (an 1 by N vector) of the last expression as we are mainly interested in understanding the second regression coefficient, $\lambda_{1,t}$:

$$\frac{1}{N} \frac{\begin{bmatrix} x_{1,t-1} - E_N[x_{i,t-1}] & \cdots & x_{N,t-1} - E_N[x_{i,t-1}] \end{bmatrix}}{Var_N[x_{i,t-1}]}$$

The final step is to multiply this 1 by N vector by the N by 1 vector R_t .

$$\lambda_{1,t} = \sum_{i=1}^N \frac{1}{N} \frac{(x_{i,t-1} - E_N[x_{i,t-1}])}{Var_N[x_{i,t-1}]} R_{i,t}$$

Define $w_{i,t-1} = \frac{1}{N} \frac{(x_{i,t-1} - E_N[x_{i,t-1}])}{Var_N[x_{i,t-1}]}$ and we have $\lambda_{1,t} = \sum_{i=1}^N w_{i,t-1} R_{i,t}$

a. Fama-MacBeth function example

```
# use Fama-MacBeth function from "linarmodels" package
```

```
y, X = dmatrices('ExRet~lnBM', StockRetAcct_DF, return_type = 'dataframe')
res1 = FamaMacBeth(y,x).fit()
FamaMacBeth(y,x).fit()
Out[6]:
```

FamaMacBeth Estimation Summary						
Dep. Variable:	ExRet	R-squared:	0.0035			
Estimator:	FamaMacBeth	R-squared (Between):	-0.0622			
No. Observations:	67745	R-squared (within):	0.0042			
Date:	Sun, Apr 04 2021	R-squared (Overall):	0.0035			
Time:	10:20:55	Log-likelihood	-4.683e+04			
Cov. Estimator:	Fama-MacBeth Standard Cov					
Entities:	8452	F-statistic:	235.61			
Avg Obs:	8.0153	P-value	0.0000			
Min Obs:	0.0000	Distribution:	$F(1, 67743)$			
Max Obs:	35.000	F-statistic (robust):	0.8958			
Time periods:	35	P-value	0.3439			
Avg Obs:	1935.6	Distribution:	$F(1, 67743)$			
Min Obs:	1601.0					
Max Obs:	2659.0					
Parameter Estimates						
	Parameter	Std. Err.	T-stat	P-value	Lower CI	Upper CI
Intercept	0.1033	0.0342	3.0213	0.0025	0.0363	0.1704
lnBM	0.0147	0.0155	0.9465	0.3439	-0.0157	0.0450

FamaMacBethResults, id: 0x23b9c3742e0

a. Fama-MacBeth using lnProf (profitability)

Only change from previous FMB code: lnProf instead of lnBM in the below

```
# the below runs regression ExRet ~ lnProf by year, including intercept
res = (StockRetAcct_DF.groupby('year').apply(ols_coef,'ExRet ~ lnProf'))
```

Mean Return:

0.1118

Profitability-sorted portfolio *IS* statistically significant
in this sample!

Std Dev:

0.2047

Sharpe Ratio

0.5461

Sharpe ratio is high

t-stat:

3.2305

a. Let's do both: Multiple Regression

```
# Let's try both lnBM and lnProf together

res = (StockRetAcct_DF.groupby('year').apply(ols_coef, 'ExRet~lnBM+lnProf'))

print('Mean Return: ', str(res[['lnBM','lnProf']].mean())+'\n',
      'Std Dev: ', str(res[['lnBM','lnProf']].std())+'\n',
      'Sharpe Ratio ', str(res[['lnBM','lnProf']].mean()/
                           res[['lnBM','lnProf']].std())+'\n',
      't-stat: ', str(35**.5*(res[['lnBM','lnProf']].mean()/
                               res[['lnBM','lnProf']].std()), sep="\n")
```

Mean Return:		
InBM	0.0194	
InProf	0.1231	B/M-sorted portfolio is slightly more statistically significant, with higher average return
Std Dev:		
InBM	0.0909	
InProf	0.2416	Profitability-sorted portfolio also has slightly higher average return than in univariate regression case
Sharpe Ratio		
InBM	0.2138	
InProf	0.5096	Both of these facts can be traced back to a negative correlation between b/m and profitability
t-stat:		
InBM	1.2646	
InProf	3.0149	

a. Let's add industry dummies

Now, both profitability and value are significant with much higher Sharpe ratios. *Why?*

```
StockRetAcct_DF['Industry']=StockRetAcct_DF['ff_ind'].astype(object)
res = (StockRetAcct_DF.groupby('year').apply(ols_coef, 'ExRet~lnBM+lnProf+Industry'))
print('Mean Returns: ', str(res.mean()[1:]), '\n',
      'Std Dev: ', str(res.std()[1:]), '\n',
      'Sharpe Ratio ', str(res.mean()[1:]/
                           res.std()[1:]), '\n',
      't-stat: ', str(35**.5*(res.mean()[1:])/
                      res.std()[1:]), sep="\n")
```

Mean Returns:	Std Dev:	Sharpe Ratio	t-stat:
Industry[T.2.0] -0.009880	Industry[T.2.0] 0.118205	Industry[T.2.0] -0.083580	Industry[T.2.0] -0.494464
Industry[T.3.0] -0.014620	Industry[T.3.0] 0.126998	Industry[T.3.0] -0.115117	Industry[T.3.0] -0.681039
Industry[T.4.0] -0.031856	Industry[T.4.0] 0.292916	Industry[T.4.0] -0.108755	Industry[T.4.0] -0.643405
Industry[T.5.0] 0.003191	Industry[T.5.0] 0.099988	Industry[T.5.0] 0.031912	Industry[T.5.0] 0.188797
Industry[T.6.0] 0.009584	Industry[T.6.0] 0.259285	Industry[T.6.0] 0.036963	Industry[T.6.0] 0.218678
Industry[T.7.0] 0.019698	Industry[T.7.0] 0.196358	Industry[T.7.0] 0.100316	Industry[T.7.0] 0.593477
Industry[T.8.0] -0.024089	Industry[T.8.0] 0.157219	Industry[T.8.0] -0.153218	Industry[T.8.0] -0.906453
Industry[T.9.0] 0.005135	Industry[T.9.0] 0.091869	Industry[T.9.0] 0.055892	Industry[T.9.0] 0.330660
Industry[T.10.0] 0.050391	Industry[T.10.0] 0.163935	Industry[T.10.0] 0.307383	Industry[T.10.0] 1.818505
Industry[T.11.0] 0.006066	Industry[T.11.0] 0.112634	Industry[T.11.0] 0.053859	Industry[T.11.0] 0.318633
Industry[T.12.0] -0.023060	Industry[T.12.0] 0.086545	Industry[T.12.0] -0.266456	Industry[T.12.0] -1.576377
InBM 0.023462	InBM 0.067146	InBM 0.349425	InBM 2.067228
InProf 0.120505	InProf 0.190310	InProf 0.633203	InProf 3.746081

a. Why did Sharpe ratios of value and profitability strategies increase?

Because, profitability and book-to-market ratios vary across industries

- Thus, an industry component in these characteristics
- But, industry exposure do not carry a risk premium (empirical statement)
 - Thus, the industry exposure just adds volatility, noise
 - So, control for this by adding industry fixed effect!

The multiple regression gets at the *marginal effect* of changing b/m and profitability

- I.e., holding industry and profitability exposures constant, what is the effect of varying the b/m ratio of the portfolio?
- I.e., holding industry and b/m exposures constant, what is the effect of varying the profitability characteristic of the portfolio?

Well, that's wonderful, but how do I trade these strategies?

a. Enhanced trading strategies

The previous Fama-MacBeth regressions give us the portfolio weights of a long-short portfolio (which leverage you can adjust as you please)

We already saw how this works in the 1-factor regression. On the previous slide, there are 14 factors (12 industry dummies (one is the intercept), $\ln BM$, and $\ln Prof$).

So, if we want to trade the marginal book-to-market effect, which is the second regressor after the intercept, we use the N_t portfolio weights given by the 2nd row of $(X'_t X_t)^{-1} X'_t$, where

$$X_t = \begin{bmatrix} 1 & \ln BM_{1,t} & \ln Prof_{1,t} & \text{indDum2}_{1,t} & \cdots & \text{indDum12}_{1,t} \\ 1 & \ln BM_{2,t} & \ln Prof_{2,t} & \text{indDum2}_{2,t} & \cdots & \text{indDum12}_{2,t} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \ln BM_{N_t,t} & \ln Prof_{N_t,t} & \text{indDum2}_{N_t,t} & \cdots & \text{indDum12}_{N_t,t} \end{bmatrix}$$

a. Enhanced trading strategies: Recipe

This exercise is what you should do when you want to pitch your new strategy!

1. Construct your trading signal (a characteristic, $z_{i,t}$)
2. Run a Fama-MacBeth regression adding other characteristics that you want to be robust to (e.g., market beta, value, profitability), and take out noise (e.g., industry exposure)
3. Get the trading strategy based on your signal by extracting the portfolio weights from the FMB regression per the previous slide. These portfolio weights ensure that the strategy (a) does not overlap with value or profitability trading and (b) is rid of unwanted noise (industry)
4. If you only want to take out noise, only add the industry dummies in the Fama-MacBeth regression (or regional dummies, like Europe, East-Asia, etc if that's what is relevant)
5. May want to run a value-weighted FMB regression to avoid too much trading (see solution to Problem Set 1)

a. Trading Strategy and Leverage

As explained earlier, the trading strategy from the Fama-MacBeth regressions yield portfolio weights that sum to zero.

- That is, the portfolios are long-short portfolios that require zero net capital (ignoring margin requirements and transaction costs)
- Also, recall that long-short portfolios are excess returns

You can thus regard this long-short portfolio as an overlay on any base portfolio you may have.

- The simplest baseline is a portfolio invested 100% in the risk-free rate
- The leverage of the portfolio can be adjusted as one pleases. As usual, changing leverage would not affect Sharpe ratios.
- Let k be a choice variable for the investor that multiplies all the Fama-MacBeth portfolio weights. Setting $k = 0$ means no exposure (obviously)
 - Then:

$$E[R_{TotPort}] = E[R_{risk-free}] + kE[R_{Fama-MacBeth}]$$

$$\sigma[R_{TotPort}] = k\sigma[R_{Fama-MacBeth}]$$

$$SR[R_{TotPort}] = \frac{E[R_{TotPort}] - E[R_{risk-free}]}{k\sigma[R_{Fama-MacBeth}]} = SR[R_{Fama-MacBeth}]$$

a. Trading Strategy and Leverage

OK, let's implement this for the value strategy, controlling for profitability and industry exposure

- I normalize the standard deviation to be 15% p.a. and compare to original simple value strategy with same standard deviation
- First show how to get portfolio weights

```
# choose the current date (end of sample)
LastDate = StockRetAcct_DF[StockRetAcct_DF['year']==2014].dropna()
LastDate = LastDate[['lnBM','lnProf']].assign(c=1).sort_index(axis=1)

# Create dummy variables for each industry. We drop the last column because we
# include a constant to avoid multicollinearity
StockRetAcct_DF['Industry']=StockRetAcct_DF['ff_ind'].astype(object)
LastDate = LastDate.join(pd.get_dummies(StockRetAcct_DF.Industry).iloc[:, :-1])

# use (X'X)^(-1) X' formula to compute weights
portweights_lnBM = np.matmul(np.linalg.inv(np.matmul(LastDate.transpose().\
values,LastDate.values)),LastDate.transpose().values)
# lnBM is second row
portweights_lnBM = np.matmul(np.array([0,1,0,0,0,0,0,0,0,0,0,0]).T, portweights_lnBM)
lnBMstdev = res.all_params.lnBM.std()
lnBMret = res.all_params.lnBM

# scale portfolio weights to get 15% standard deviation of returns
portweights_lnBM = portweights_lnBM*0.15/lnBMstdev
```

a. Trading Strategy and Leverage

Next, let's do this in an out-of-sample fashion through the whole sample and plot

```
y, X = dmatrices('ExRet~lnBM+lnProf+C(Ind)', StockRetAcct_DF, return_type = 'dataframe')
res = FamaMacBeth(y,X).fit()
lnBMstd = res.all_params.lnBM.std()
lnBMret = res.all_params.lnBM
# for plotting, get the scaled excess portfolio returns
lnBM_ret = lnBMret*0.15/lnBMstd

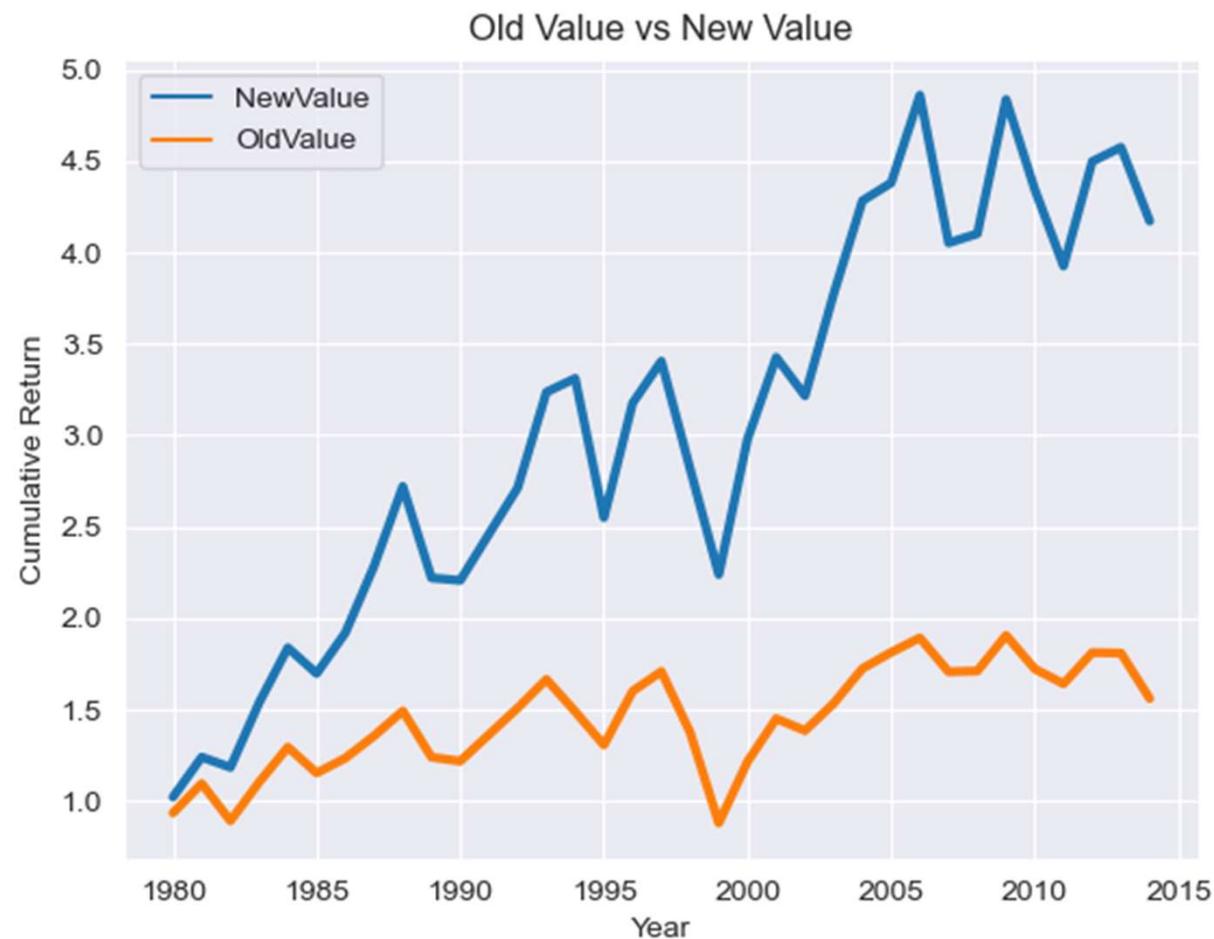
# create cumulative log return series
cum_ret_lnBM = pd.DataFrame.cumsum(np.log(1+lnBM_ret))

# get "old" simple value strategy returns
lnBMstd = res1.all_params.lnBM.std()
lnBMret = res1.all_params.lnBM
lnBM_old_ret = pd.DataFrame.cumsum(np.log(1+lnBMret*0.15/lnBMstd))

summary = pd.DataFrame()
summary['NewValue'] = np.exp(cum_ret_lnBM)
summary['OldValue'] = np.exp(lnBM_old_ret)

# Plot Old Value vs New Value
sns.set_style('darkgrid')
plt.figure()
ax=sns.lineplot(data=summary,dashes = False,linewidth = 3)
ax.set(xlabel = 'Year',
       ylabel = 'Cumulative Return',
       title = "Old Value vs New Value")
```

a. Old Value vs. New (improved) Value



a. MR (multiple regression) vs. SR (simple regression)

Another verbalization of what we just discussed:

As book-to-market increases across firms, there is a “pure” effect of increasing discount rates (higher future returns).

But there are also strong industry component in book-to-market. Industry risk has historically not been priced (marginally).

Thus, the book-to-market coefficient in the simple regression reflects two effects:

1. “direct” effect which is the return predictor and it is positive
2. “indirect” effects from industry and profitability which are both correlated with book-to-market

The Simple Regression coefficient is the sum of these effects which is smaller than the “pure” book-to-market effect.

b. Omitted Variable Bias

Econometricians call what we just saw “omitted” variables bias.

Suppose the “true” effect of a variable X is from a regression of Y on X,Z.

$$y_i = \alpha + \beta x_i + \gamma z_i + \varepsilon_i$$

But we actually only run Y on X (that is, we “omit” Z). Then the least squares coefficient on X will no longer be a pure estimate even if we have an infinite amount of data.

$$y_i = \alpha + \beta^* x_i + \varepsilon_i^*$$

b. Omitted Variable Bias

For large samples the difference between the univariate regression estimate and the multivariate regression estimate will be approximately

$$\hat{\beta}_{SR} - \hat{\beta}_{MR} \approx \gamma \frac{\text{cov}(X, Z)}{\text{var}(Z)}$$

This quantity is often called the “omitted variable bias.” Obviously, there is no omitted variable bias when Z is not correlated with X. The larger the **indirect effect** of Z on X, the larger the bias.

BUT, even if Z has a large effect on Y, omitted variable bias is only present if X and Z are correlated!!!

c. Panel regression overview

The simplest setting for “*big data*” type analysis are panel regressions.

- Panel: Typically cross-section *and* time series, size $N \times T$
 - **Balanced panel:** There are N observations in cross-section for each t
 - **Unbalanced panel:** For each t only a subset of the cross-section have data ($N(t) < N$)

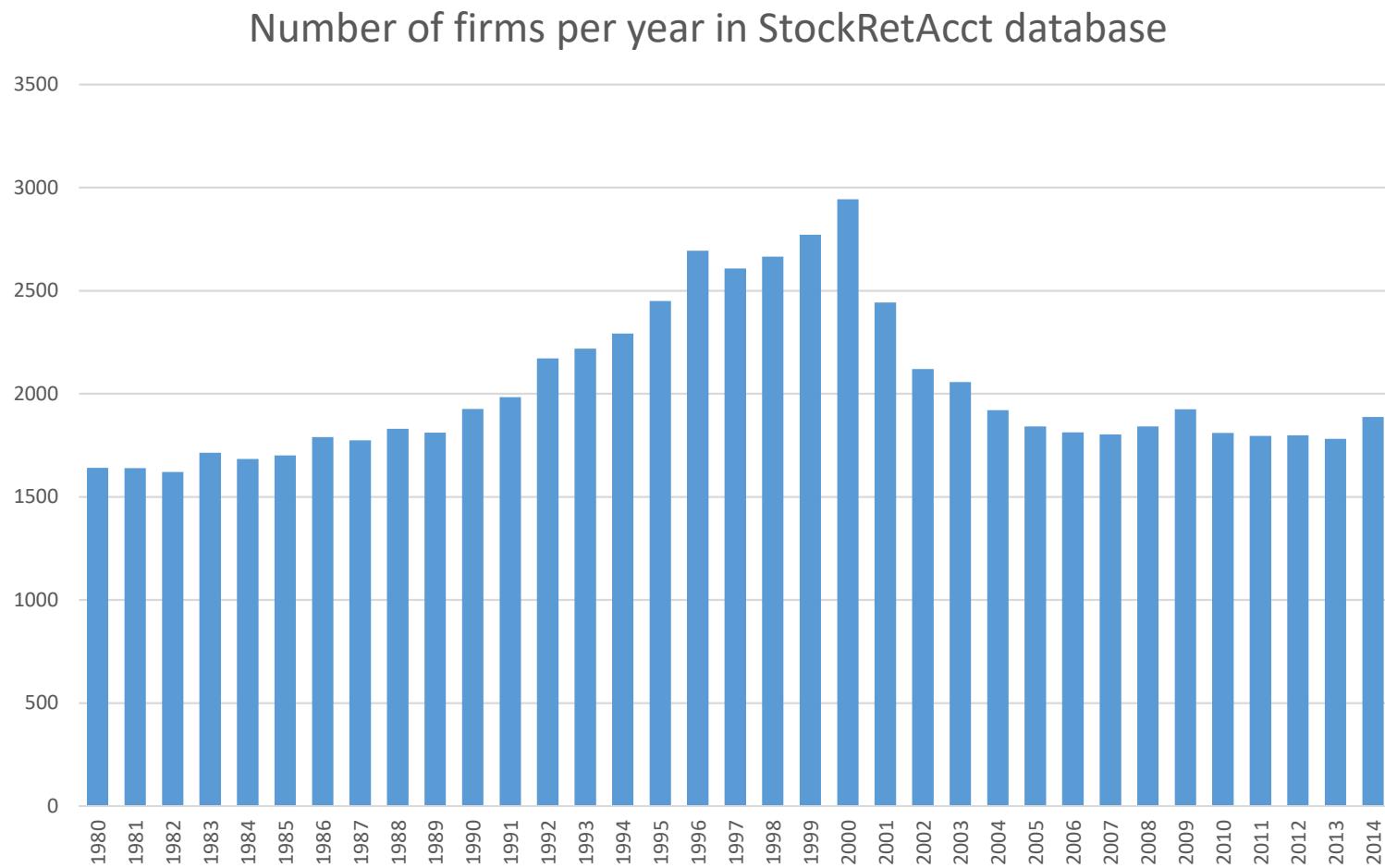
The dataset StockRetAcct_DT.csv that we have been working with is an example of an unbalanced panel

- Total set of firms, N , are 20,314.
 - At each time t , there are way fewer firms “alive”
 - See plot on next slide
- 35 years of observations, so $T = 35$

While *Machine Learning* often refers to nonlinear techniques to analyze data, ***your first analysis will (and should!) typically be linear regressions***

- Robust, easy-to-understand and -communicate

c. Cross-section of firms is an unbalanced panel



c. Panel regression overview

Panel uses variation both in cross-section and over time to identify regression coefficients. That's BIG DATA!

- **Implicit assumption: slope coefficients do not vary over time or across firms**
- However, intercept may be allowed to vary:
 - Over time: **Time fixed effects**
 - Across firms: **Firm fixed effects**

Canonical panel regression:

- delta's are *time fixed effects*, theta's are *firm fixed effects*. No i subscript on beta
- Note that the error term may be correlated across firms and time, so standard errors typically cannot be found using the classic OLS standard error formula that presumes iid error terms.

$$y_{i,t} = \delta_t + \theta_i + \beta' X_{i,t} + \varepsilon_{i,t}$$

c. Panel regression: increasing power

A simple example of how panel (pooled) regressions can increase power

- Assume all stocks have the same expected return and return variance = σ^2
- Assume all pairwise *correlations* equal ρ **across stocks, zero over time**
- Simplest model: X's is a vector of ones, i.e., just estimating intercept

$$R_{i,t} = \mu + \varepsilon_{i,t}$$

The estimate is simply $\hat{\mu} = \frac{1}{NT} \sum_{t=1}^T \sum_{i=1}^N R_{i,t}$

The variance of this estimate is

$$\text{var}(\hat{\mu}) = E \left[\left(\frac{1}{NT} \sum_{t=1}^T \sum_{i=1}^N R_{i,t} - \mu \right)^2 \right] = \frac{\sigma^2}{NT} (1 + (N-1)\rho)$$

Notice how if stocks are not perfectly correlated, the variance of the estimate is lower than if we estimated each stock's mean return separately (which has a variance of estimate of σ^2/T)

c. Panel regression: simplest example

- Consider the below simplest panel regression (one regressor plus intercept)

$$y_{i,t} = \alpha + \beta x_{i,t} + \varepsilon_{i,t}$$

- Thus, there are no firm, industry, or year fixed effects.
- Assume a balanced panel where $i = 1, \dots, N; t = 1, \dots, T$.
- How do you estimate this (other than using the *Python* routine)?

c. Panel regression: simplest example

- Define the following matrices:

$$\begin{array}{c} X_{\tilde{\omega}} \\ TN \times 2 \end{array} = \begin{bmatrix} 1 & x_{1,1} \\ 1 & x_{1,2} \\ 1 & \vdots \\ 1 & x_{1,T} \\ 1 & x_{2,1} \\ 1 & \vdots \\ 1 & x_{2,T} \\ 1 & \vdots \\ 1 & x_{N,1} \\ 1 & x_{N,2} \\ 1 & \vdots \\ 1 & x_{N,T} \end{bmatrix} \quad \begin{array}{c} Y_{\tilde{\omega}} \\ TN \times 1 \end{array} = \begin{bmatrix} y_{1,1} \\ y_{1,2} \\ \vdots \\ y_{1,T} \\ y_{2,1} \\ \vdots \\ y_{2,T} \\ \vdots \\ y_{N,1} \\ y_{N,2} \\ \vdots \\ y_{N,T} \end{bmatrix}$$

c. Panel regression: simplest example

- Then, we find the regression coefficients as:

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} = (X'X)^{-1}X'Y$$

- Note, however, that *assuming errors are i.i.d. as in standard OLS is not a good idea here. Almost surely, firms' residuals are cross-sectionally correlated* (e.g., a shock to the state of the economy affects all firms' residuals).
- There could also be autocorrelation over time for each firm
- The variance-covariance matrix of the residuals is huge however:
 - TN x TN
 - In order to decrease the size of the covariance matrix (in terms of estimating it), we typically apply *clustering* (see next slide)

c. Clustering

- Assume that firms' shocks are correlated within each year but not across years.
- Assume also that the cross-firm covariance is constant over time.
 - That is: $\text{Cov}(\varepsilon_{i,t}, \varepsilon_{j,t+k}) = \sigma_{ij}$ for all t if $k = 0$ but zero if $k \neq 0$.
- Clustering standard errors by **time** (e.g., year) imposes these assumptions
 - Note that we now have “only” $N(N+1)/2$ free coefficients in the variance-covariance matrix
 - However, while still a lot coefficients, we are looking for specific averages that makes up the covariance matrix of the two estimated coefficients (which is only a 2 by 2 matrix), so estimation error in the still big covariance matrix of residuals washes out to a large extent. (See paper by Mitchell Petersen on BruinLearn if interested in further details).
- Now, assume firm residuals also can be autocorrelated, though only within-firm, not across firms.
 - That is: $\text{Cov}(\varepsilon_{i,t}, \varepsilon_{i,t+k}) = \sigma_{iik}$ for all t and $\text{Cov}(\varepsilon_{i,t}, \varepsilon_{j,t+k}) = 0$ for $i \neq j$ and $k \neq 0$.
 - Adding clustering by **firm** achieves this.
- Allowing for heteroskedasticity in addition to clustering can be achieved using “Rogers” standard errors.

c. Fixed effects prelude

- Before getting into fixed effects, consider the standard regression

$$y_{i,t} = \alpha + \beta x_{i,t} + \varepsilon_{i,t}$$

- Recall, from standard OLS results, that if we define

$$\tilde{x}_{i,t} \equiv x_{i,t} - \frac{1}{NT} \sum_{t=1}^T \sum_{i=1}^N x_{i,t} \quad \text{and} \quad \tilde{y}_{i,t} \equiv y_{i,t} - \frac{1}{NT} \sum_{t=1}^T \sum_{i=1}^N y_{i,t}$$

the β and $\varepsilon_{i,t}$ in the below regression are identical to the ones at the top

$$\tilde{y}_{i,t} = \beta \tilde{x}_{i,t} + \varepsilon_{i,t}$$

- **In sum, the intercept is ‘taking out the means’**

c. Fixed effects

- Now, allow for firm fixed effects:

$$y_{i,t} = \alpha_i + \beta x_{i,t} + \varepsilon_{i,t}$$

- For simplicity, assume $N = 2$.
- Then Y is the same as before, but X has firm dummies and becomes:

$$\underbrace{\begin{matrix} X \\ TN \times 3 \end{matrix}}_{\text{TN} \times 3} = \begin{bmatrix} 1 & 0 & x_{1,1} \\ 1 & 0 & x_{1,2} \\ \vdots & \vdots & \vdots \\ 1 & 0 & x_{1,T} \\ 0 & 1 & x_{2,1} \\ 0 & 1 & x_{2,2} \\ \vdots & \vdots & \vdots \\ 0 & 1 & x_{2,T} \end{bmatrix} \quad \text{and we have} \quad \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \beta \end{bmatrix} = (X'X)^{-1}X'Y$$

c. Fixed effects

- What does this firm fixed effect achieve?
- Calculate beta explicitly using the usual $(X'X)^{-1}X'Y$ formula

$$\beta = \frac{Cov(y_{i,t} - \bar{y}_i, x_{i,t} - \bar{x}_i)}{Var(x_{i,t} - \bar{x}_i)} \quad \text{where } \bar{y}_i \equiv \frac{1}{T} \sum_{t=1}^T y_{i,t}, \quad \bar{x}_i \equiv \frac{1}{T} \sum_{t=1}^T x_{i,t}$$

- In words: with firm fixed effects, we are capturing variation in the independent and dependent variable as deviations from each firms' mean of these variables, not from the grand mean as we would in a typical regression
- It is as if we are demeaning all variables at the firm-level. I.e., we are not explaining variation in means across firms with beta.
- Below, for comparison, is the beta from a regression **without** the fixed effect:

$$\beta = \frac{Cov(y_{i,t} - \bar{y}, x_{i,t} - \bar{x})}{Var(x_{i,t} - \bar{x})} \quad \text{where } \bar{y} \equiv \frac{1}{TN} \sum_{t=1}^T \sum_{i=1}^N y_{i,t}, \quad \bar{x} \equiv \frac{1}{TN} \sum_{t=1}^T \sum_{i=1}^N x_{i,t}$$

c. Panel regression and Fama-MacBeth

The Fama-MacBeth looks a lot like the panel regression on the previous slides

- *After all: we are trying to estimate, for example, δ_0 and δ_1 in:*

$$R_{i,t+1} = \delta_0 + \delta_1 \ln BM_{i,t} + \varepsilon_{i,t+1}$$

However:

- While Fama-MacBeth is a kind of panel approach, it cross-sectionally demeans and effectively standardizes the predictor variable at each time t , and in addition weighs each time t coefficient the same when taking the average
 - Thus, portfolio returns from years where there are few firms are weighted as much as portfolio returns from years where there are many firms
 - A standard panel regression does not do this. It weighs each observation equally (unless you specify a weighted panel regression).

d. Panel regressions: case study

We will use the function *panelOLS* from package “*linearmodels*”

Our application will be ***firm-level earnings forecasting***

- Idea: using large set of historical data may beat analyst forecasts
 - Actually, analysts have been shown to have a strong upwards bias in their earnings forecasts and poor overall ability to be better than very simple models
- Forecasting earnings or other cash flow related quantities is important for valuations (possible input to trading strategies), as well as capital budgeting within firms
- Methodology is general – could also be used for forecasting realized variances, covariances, sales, ‘clicks’, basically anything in a panel setting...

d. Firm Earnings

Our dependent variable will be *Return on Equity* (ROE):

$$ROE_{i,t} = \frac{Net\ Income_{i,t}}{Book\ Equity_{i,t-1}}$$

We will continue using the “StockRetAcct”-dataset, which contains many accounting variables as well as returns and market values.

In particular, we will consider the variable *InROE*, which is the log of 1 + ROE minus log inflation (I.e., log real ROE).

For now, we will ignore out-of-sample vs. in-sample issues

d. Firm Earnings

First, some summary statistics (across firms and time):

1. Mean lnROE = 8.8%
2. St.dev. lnROE = 27.4%

Consider the simple panel forecasting model:

$$\ln ROE_{i,t+1} = \delta + \beta' X_{i,t} + \varepsilon_{i,t+1}$$

Let's start with lagged lnROE as the forecasting variable (the obvious one).

d. Firm Earnings: Panel Forecasting

Regression with no FE or SE

```
roe_panel = PanelOLS.from_formula(formula='lead_lnROE ~ 1+lnROE', data=StockRetAcct_DF).fit()
print(roe_panel)
```

PanelOLS Estimation Summary

Dep. Variable:	lead_lnROE	R-squared:	0.2734
Estimator:	PanelOLS	R-squared (Between):	0.7424
No. Observations:	60467	R-squared (Within):	0.2549
Date:	Sun, Apr 04 2021	R-squared (Overall):	0.2734
Time:	12:17:15	Log-likelihood	1.116e+04
Cov. Estimator:	Unadjusted	F-statistic:	2.275e+04
Entities:	12	P-value	0.0000
Avg Obs:	5038.9	Distribution:	F(1,60465)
Min Obs:	1658.0		
Max Obs:	1.064e+04	F-statistic (robust):	2.275e+04
Time periods:	34	P-value	0.0000
Avg Obs:	1778.4	Distribution:	F(1,60465)
Min Obs:	1497.0		
Max Obs:	2302.0		

Decent predictability: R² = 27%

Positive autocorrelation

Careful about those t-statistics!

- Created under (bad) assumption of i.i.d. error terms

Parameter Estimates

	Parameter	Std. Err.	T-stat	P-value	Lower CI	Upper CI
Intercept	0.0408	0.0009	45.853	0.0000	0.0390	0.0425
lnROE	0.5224	0.0035	150.83	0.0000	0.5156	0.5292

d. Standard Errors: Clustering

Firms' shocks to earnings are correlated within firms

- E.g., a firm's positive earnings shock is autocorrelated
 - To account for this add standard error clustering at the firm level
 - Set index for 'entity' and 'time' so panel regression can adjust along these dimensions

Clustering!

```
# Regression with no FE, standard errors clustered at the firm level
roe_panel2 = PanelOLS.from_formula(formula='lead_lnROE ~ 1+lnROE',
                                    data=StockRetAcct_DF).fit(cov_type = 'clustered',
                                                               cluster_entity=True, cluster_time=False)
```

Parameter Estimates

	Parameter	Std. Err.	T-stat	P-value	Lower CI	Upper CI
Intercept	0.0408	0.0021	19.644	0.0000	0.0367	0.0448
lnROE	0.5224	0.0148	35.357	0.0000	0.4935	0.5514

Note: R2 didn't change

- But, t-stats much smaller

d. Standard Errors: Clustering

- Cluster also on time (double-clustering)
- Simplest example: economy enters expansion, all firms make more money

```
# Regression with no FE, standard errors clustered at the firm and year level
roe_panel3 = PanelOLS.from_formula(formula='lead_lnROE ~ 1+lnROE',
                                    data      = StockRetAcct_DF).fit(cov_type = 'clustered',
                                    cluster_entity=True, cluster_time=True)
```

Clustering! (on firm and year)



Parameter Estimates

	Parameter	Std. Err.	T-stat	P-value	Lower CI	Upper CI
Intercept	0.0408	0.0057	7.0945	0.0000	0.0295	0.0520
lnROE	0.5224	0.0320	16.309	0.0000	0.4596	0.5852

Note: Again R2 and coefficients didn't change (nor should they)

- But, t-stats even smaller, evidence of both firm and time dependencies in the errors

d. Fixed Effects

Well-known permanent industry effects in accounting variables

- Due to different production opportunities as well as conventions
- Add an industry fixed effect (δ_j), like adding industry dummies:

$$\ln ROE_{i,j,t+1} = \delta_j + \beta' X_{i,t} + \varepsilon_{i,t+1}$$

Here j denotes industry and i is firm.

Be careful using firm-level fixed effects!! (That's like adding a dummy variable for each firm)

- Small sample issues are severe since median firm life in sample is only 10 years
- Thus, firm-level average is badly estimated, which affects other regression coefficients as well.
- Try to avoid. If you can't, run extensive Monte-Carlo simulations to assess likely magnitude of biases

d. Fixed Effects

```
# Create lead lnROE for each firm
StockRetAcct_DF.sort_values(['FirmID', 'year'], ascending=[True, True], inplace=True)
StockRetAcct_DF['lead_lnROE'] = StockRetAcct_DF.groupby('FirmID')['lnROE'].\  
.
```

```
# Regression with industry FE, standard errors clustered at the year level
roe_panel4 = PanelOLS.from_formula(formula='lead_lnROE~1+lnROE+C(ff_ind)',  
                                    data = StockRetAcct_DF).fit(cov_type = 'clustered',  
                                    cluster_entity=True, cluster_time=True)
```

PanelOLS Estimation Summary

Dep. Variable:	lead_lnROE	R-squared:	0.2823
Estimator:	PanelOLS	R-squared (Between):	0.4345
No. Observations:	60467	R-squared (Within):	0.0276
Date:	Sun, Apr 04 2021	R-squared (Overall):	0.2823
Time:	13:56:13	Log-likelihood	1.153e+04
Cov. Estimator:	clustered	F-statistic:	1981.1
Entities:	6661	P-value	0.0000
Avg Obs:	9.0778	Distribution:	F(12, 60454)
Min Obs:	1.0000	F-statistic (robust):	67.955
Max Obs:	34.000	P-value	0.0000
Time periods:	34	Distribution:	F(12, 60454)
Avg Obs:	1778.4		
Min Obs:	1497.0		
Max Obs:	2302.0		

Parameter Estimates

	Parameter	Std. Err.	T-stat	P-value	Lower CI	Upper CI
lnROE	0.5046	0.0311	16.201	0.0000	0.4436	0.5657

Fixed effect, at the industry level as defined by the ff_ind variable

Now, the R2 increased slightly

- Note: industry dummies are not reported in output
- Not a huge effect in this case.

d. Fixed Effects

Let's also add year fixed effects:

$$\ln ROE_{i,j,t+1} = \delta_j + \gamma_t + \beta' X_{i,t} + \varepsilon_{i,t+1}$$

Interpretation

- Remove average response (market factor)
- For instance: “I am not interested in trying to predict the market-level movements in earnings, just out- or under-performance relative to market”

d. Fixed Effects

```
roe_panel5 = PanelOLS.from_formula(formula=\n    'lead_lnROE ~ 1+lnROE+C(ff_ind)+TimeEffects',\n    data      = StockRetAcct_DF).fit(cov_type = 'clustered',\n                                      cluster_entity=True, cluster_time=True)
```

PanelOLS Estimation Summary

		R-squared:	0.2793
Dep. Variable:	lead_lnROE	R-squared (Between):	0.4341
Estimator:	PanelOLS	R-squared (Within):	0.0280
No. Observations:	60467	R-squared (Overall):	0.2822
Date:	Sun, Apr 04 2021	Log-likelihood	1.189e+04
Time:	14:20:31	F-statistic:	1951.7
Cov. Estimator:	Clustered	P-value	0.0000
Entities:	6661	Distribution:	F(12, 60421)
Avg Obs:	9.0778	F-statistic (robust):	69.835
Min Obs:	1.0000	P-value	0.0000
Max Obs:	34.000	Distribution:	F(12, 60421)
Time periods:	34		
Avg Obs:	1778.4		
Min Obs:	1497.0		
Max Obs:	2302.0		

Parameter Estimates

	Parameter	Std. Err.	T-stat	P-value	Lower CI	Upper CI
lnROE	0.5036	0.0316	15.917	0.0000	0.4416	0.5656

TimeEffects refers to dummies of time-variable in key (EntityEffects would put in fixed effects for FirmID)

Now, the R2 again increased slightly

- But, not a large effect. Thus, variation in earnings is mainly cross-sectional

d. Predicting firm earnings, big model

```
roe_panel6 = PanelOLS.from_formula(formula=\
    'lead_lnROE~1+lnROE+lnBM+lnProf+lnLever+lnIssue+lnInv+C(ff_ind)',\
        data      = StockRetAcct_DF).fit(cov_type = 'clustered',\
            cluster_entity=True, cluster_time=True)
```

Parameter Estimates

	Parameter	Std. Err.	T-stat	P-value	Lower CI	Upper CI
Intercept	-0.0024	0.0083	-0.2917	0.7705	-0.0186	0.0138
C(ff_ind)[T.2.0]	-0.0080	0.0058	-1.3830	0.1667	-0.0193	0.0033
C(ff_ind)[T.3.0]	4.238e-05	0.0041	0.0103	0.9918	-0.0080	0.0081
C(ff_ind)[T.4.0]	-0.0145	0.0079	-1.8301	0.0672	-0.0301	0.0010
C(ff_ind)[T.5.0]	-0.0025	0.0049	-0.5139	0.6073	-0.0120	0.0070
C(ff_ind)[T.6.0]	-0.0371	0.0072	-5.1441	0.0000	-0.0512	-0.0230
C(ff_ind)[T.7.0]	-0.0637	0.0118	-5.4120	0.0000	-0.0867	-0.0406
C(ff_ind)[T.8.0]	0.0317	0.0053	5.9252	0.0000	0.0212	0.0422
C(ff_ind)[T.9.0]	-0.0054	0.0033	-1.6448	0.1000	-0.0119	0.0010
C(ff_ind)[T.10.0]	-0.0457	0.0066	-6.8942	0.0000	-0.0587	-0.0327
C(ff_ind)[T.11.0]	0.0388	0.0080	4.8232	0.0000	0.0230	0.0546
C(ff_ind)[T.12.0]	-0.0104	0.0044	-2.3359	0.0195	-0.0191	-0.0017
lnROE	0.3621	0.0222	16.290	0.0000	0.3186	0.4057
lnBM	-0.0535	0.0057	-9.4403	0.0000	-0.0646	-0.0424
lnProf	0.2295	0.0293	7.8363	0.0000	0.1721	0.2869
lnLever	-0.0100	0.0046	-2.1632	0.0305	-0.0191	-0.0009
lnIssue	-0.0623	0.0079	-7.8838	0.0000	-0.0778	-0.0468
lnInv	-0.0559	0.0140	-3.9926	0.0001	-0.0834	-0.0285

Now, the R2 again increased

- Many variables are significant
- Note: marginal effect of higher investment is lower future ROE...!

d. Predicting earnings in 5 years, big model

```
# What predicts ROE in five years? First, create a five year lagged column
StockRetAcct_DF['lead_lnROE5'] = StockRetAcct_DF.groupby('FirmID')['lnROE'].shift(-5)

# Industry component in accounting variables, include industry FE,
# standard errors clustered at the firm and year level
roe_panel7 = PanelOLS.from_formula(formula=\
    'lead_lnROE5~1+lnROE+lnBM+lnProf+lnLever+lnIssue+lnInv+c(ff_ind)',\
    data      = StockRetAcct_DF).fit(cov_type = 'clustered',\
    cluster_entity=True, cluster_time=True)
```

PanelOLS Estimation Summary

Dep. Variable:	lead_lnROE5	R-squared:	0.0877			
Parameter Estimates						
	Parameter	Std. Err.	T-stat	P-value	Lower CI	Upper CI
lnROE	0.0681	0.0196	3.4788	0.0005	0.0297	0.1064
lnBM	-0.0252	0.0047	-5.4083	0.0000	-0.0344	-0.0161
lnProf	0.1681	0.0346	4.8617	0.0000	0.1003	0.2358
lnLever	0.0019	0.0041	0.4642	0.6425	-0.0062	0.0100
lnIssue	-0.0796	0.0099	-8.0276	0.0000	-0.0990	-0.0602
lnInv	-0.0526	0.0096	-5.4790	0.0000	-0.0714	-0.0338

Now, the R2 is quite a bit lower, as we would expect

- Still, many variables are significant
- Note: marginal effect of higher investment is still lower future ROE...!

d. Earnings prediction model

Used over 50,000 firm-year observations to create earnings forecasting model

- In-sample R² was around 35%
- About 9% when predicting the annual earnings 5 years from now

Benefit a lot from cross-section

- Year fixed effects not that important
- Most variation is cross-sectional, that's how we get such high *t*-stats
- Can use model for a new firm even
 - Set all characteristics you do not have to their unconditional average
- Impossible to run individual model at the firm-level given only 10 year median firm survival in data

d. Firm variance prediction model

We can use the same model to predict firm variance.

In fact, that is what you will do in Problem Set 2

e. Panel regression postscript

Main idea:

- **Get power** from looking at both time-series and cross-section
- Assumes ‘betas’ are the same across time and firms
- Can remove time and firm (or industry) fixed effects
 - Time f.e.: Identification of beta entirely from cross-sectional variation in response (y) from feature (x)
 - Firm f.e.: Identification of beta entirely from time-series variation in response (y) from feature (x)
- Need to make sure standard errors are appropriate to account for potential correlation patterns (across T and N; clustering)
- Routines exist for both unbalanced and balanced panels.
- Big data: time is typically short, cross-section can be huge!

e. Fama-MacBeth and Panel: When to use

A cross-sectional (Fama-MacBeth) regression is a convenient way to do portfolio sorts!

- Especially powerful when we have multiple right-hand side variables
- Realized excess return on portfolio k is the regression coefficient, $\lambda_{t,k}$
- Risk premium on each portfolio is then estimated using the portfolio's average sample return
 - This estimate will be quite noisy, unless you have a really long time-series sample

Panel regression estimates (constant) regression coefficients using both time-series and cross-sectional variation

- Big increase in power (regression coefficients have low standard error)
- However, requires assumption that regression coefficients are constant over time and in cross-section (potentially with exception of intercepts)

e. Fama-MacBeth and Panel: When to use

Year fixed effects are appropriate if all you are interested in are cross-sectional differences

- E.g., predict the difference of two stocks' returns or earnings

Industry-fixed effects are appropriate if you believe each industry has permanent differences in the y variable (e.g., one can argue this is the case for ROE).

- If there are many firms in each industry, these industry fixed effects can be estimated with relatively little noise

Firm fixed effects I do not advise in forecasting exercises. These typically lead to overfitting and small-sample biases that can be quite severe.

- A cross-validation exercise, which we discuss later (Topic 4), will typically show that this is indeed the case and lead you to drop firm fixed-effects in your prediction exercise