Session 5 Modified Jones Model

Data and Stata Code

All the Stata code can be found in the do-file.

Updated: You may do the following to reduce the running time of the loops:

1. First define the variables we need (i.e. ta, x1, x2, x3) and drop observations with the missing values, then define the boarders of the loops (see the updated order of code below).
2. Divide the datasets into multiple sub-datasets basing on industry (SIC). e.g. keep only the SIC = 01, save as a separated file; then define the boarders, and run the regression.
3. You may run regression on these sub-datasets on different machines simultaneously.

Get data from Compustat

WRDS -> Compustat -> North America-Daily -> Fundamentals Annual

Step 1: Date range: 10 years (eg. 2009-1 to 2018-12)

Step 2: Search the entire database

Step 3: Variables: GVKEY, SIC, CUSIP, IB, OANCF, REVT, RECT, AT, PPEGT

Step 4: .dta

Save the dataset as Compustat.dta.

Clean the raw data

Drop the observations with industry format = FS, missing total asset, or missing SIC.

```
. drop if indfmt =="FS"
(11,723 observations deleted)

. drop if at ==.
(27,335 observations deleted)

. drop if sic ==""
(0 observations deleted)
```

Check the duplication.

```
. duplicates tag fyear gvkey , gen (dup)

Duplicates in terms of fyear gvkey

. tab dup
```

dup	Freq.	Percent	Cum.
0	86,628	100.00	100.00
Total	86,628	100.00	

Convert the variable type of SIC and gvkey.

```
. tostring sic, replace
sic already string; no replace

. destring gvkey, replace
gvkey: all characters numeric; replaced as long
```

Generate the 2-digit SIC (i.e. keep the first 2 digit), and convert it to numeric type.

```
. gen sic_2 = substr(sic, 1, 2)

. destring sic_2, replace
sic_2: all characters numeric; replaced as byte
```

Prepare the dataset for regression

Generate a blank variables (uhat) to store the residuals from the later regressions.

```
. gen uhat =.
(86,628 missing values generated)
```

Declare the panel data.

```
. xtset gvkey fyear
       panel variable:  gvkey (unbalanced)
        time variable:  fyear, 2008 to 2018, but with gaps
                delta:  1 unit
```

[Please notice the order of the code has changed]

Generate variables according to the Modified Jones Model:

Modified Jones model (Dechow et al., 1995)

$$Acc_t = \alpha + \beta_1(\Delta Rev_t - \Delta Rec_t) + \beta_2 PPE_t + \varepsilon_t$$

- ta: total accrual = (Income Before Extraordinary Item – Operating Net Cash Flow) / Total Asset (Lagged)

```
. gen ta = (ib - oancf ) / L.at
(14,405 missing values generated)
```

- x1: intercept scaled by Total Asset (Lagged)

```
. gen x1 = 1/L.at
(13,962 missing values generated)
```

- x2: (ΔRev – ΔRec) scaled by Total Asset (Lagged)

```
. gen x2 = (d.revt - d.rect )/L.at
(15,421 missing values generated)
```

- x3: PPE (gross) scaled by Total Asset (Lagged)

```
. gen x3 = ppegt / L.at
(23,658 missing values generated)
```

Drop observations with missing values.

```
. drop if ta ==.
(14,405 observations deleted)

. drop if x1 ==.
(0 observations deleted)

. drop if x2 ==.
(1,378 observations deleted)

. drop if x3 ==.
(9,315 observations deleted)
```

[Please notice the order of the code has changed]

Find and hold the boarders of SIC and fyear.

```
. sum sic_2
```

Variable	Obs	Mean	Std. Dev.	Min	Max
sic_2	61,530	42.77887	22.32319	1	99

```
. scalar a= r(min)

. scalar b= r(max)
```

```
. sum fyear
```

Variable	Obs	Mean	Std. Dev.	Min	Max
fyear	61,530	2013.833	2.567118	2009	2018

```
. scalar c= r(min)

. scalar d= r(max)
```

Generate variable obs representing rows, find and hold boarder of obs.

```
. gen obs= [_n]

. sum obs
```

Variable	Obs	Mean	Std. Dev.	Min	Max
obs	61,530	30765.5	17762.33	1	61530

```
. scalar e= r(min)

. scalar f= r(max)
```

Regression

Run regression for each industry-year.

Loop in Industry (i: industry)

Loop in Year (x: year)

Loop in Firm (j: observation)

```
. forvalues i= `=scalar(a)'/`=scalar(b)' {
  2. forvalues x= `=scalar(c)'/`=scalar(d)' {
  3. forvalues j= `=scalar(e)'/`=scalar(f)' {
  4. capture noisily reg ta x1 x2 x3 if sic_2==`i' & fyear==`x' & obs != `j', nocons
  5. capture noisily predict uhat_2, resid
  6. capture noisily replace uhat_2=. if e(N) <10
  7. capture noisily replace uhat= uhat_2 if sic_2==`i' & fyear==`x' & obs==`j'
  8. capture noisily drop uhat_2
  9. di `i'
 10. di `x'
 11. di `j'
 12. }
 13. }
 14. }
```

```
1. forvalues i= `=scalar(a)'/`=scalar(b)' {
```

- Loop: for every i (representing industry) in boarder [a, b] we defined in the previous step

```
2. forvalues x= `=scalar(c)'/`=scalar(d)' {
```

- Loop: for every x (representing year) in boarder [c, d] we defined in the previous step

```
3. forvalues j= `=scalar(e)'/`=scalar(f)' {
```

- Loop: for every j (representing observation rows) in boarder [e, f] we defined in the previous step

```
4. capture noisily reg ta x1 x2 x3 if sic_2==`i' & fyear==`x' &
obs != `j', nocons
```

- Run regression for each industry-year, and exclude the firm itself when regressing.
- Capture: execute command without output; allows the program to continue despite errors
  - noisily: display the output and any error messages

```
5. capture noisily predict uhat_2, resid
```

- Predict the residual of each regression and store it in uhat_2.

```
6. capture noisily replace uhat_2=. if e(N) < 10
```

- Drop the uhat_2 (replace as missing) if there is fewer than 10 observations.

```
7. capture noisily replace uhat= uhat_2 if sic_2==`i' &
fyear==`x' & obs== `j'
```

- Assign the value of uhat_2 to uhat of each observation.
- uhat will be the indicator of discretionary accrual (i.e. abnormal accrual).

```
8. capture noisily drop uhat_2
```

- Drop the uhat_2 to get ready for the next loop.

```
9. di `i'
10. di `x'
11. di `j'
```

- Display the current i, x, and j.

```
12. }
13. }
14. }
```

Note: Be careful and patient when you execute the loops.