NotesS6 Inital Regression Output

Andrew Wang

2024-03-11

The code below is intended to create the dependent and indepdent variables in the transactional dataset, and the perform the fixed effects regression on the dataset. In order to create the independent variable, we had to bring forward calculations from step 5 regarding entry and exit, and then merge this back into the transactions dataset by joining on fund_id and transaction quarter. Then, this data was lagged to produce lagged entry values.

There were a few assumptions made in this methodology that may need to be changed, they are outlined below: \ - We do not omit any transactional data from regardless of the value of h, which means for something like h = 8, all obs are null before Q4 2006

- Act4Q is zero until Q4 2005, as lag produces zeroes. This means for all fund_ids with some volume over the first four quarters, we have entry = 1 (fund is an entrant) for Q4 2004 - Q3 2005
- An investor was considered on the basis of fund_id

Note that in the interest of performance, much of the methodology code is commented out and instead the data is directly read in. This is to avoid having to compute all grouped lag operations again and instead just use a previous stored version of the computations

```
#Add dep var dummy
df$epats <- ifelse(df$elec_platf == 1, 1, 0)
df$epats <- ifelse(df$ATS == 1, 1, df$epats)

# #Add indep var dummy
# #Need to create al entry exit points, then join on combinations of fund_id and yearqtr to pull forwar
# ###Pull forward methodology for identifying lag from step 5
#
# ###---------------------------------[Step 5]-------------------------------------###
# s5 <- df %>% group_by(Time, fund_id) %>% transmute(
#   quartVol = sum(trns_amount)
# ) %>% as.data.frame %>% unique()
# s5$Time <- as.POSIXct(s5$Time)
#
# #Add all pairwise combinations of rows
# dates <- unique(s5$Time)
# fund_ids <- unique(s5$fund_id)
# pairs <- expand.grid(Time = dates, fund_id = fund_ids, stringsAsFactors = F)
# pairs$quartVol <- 0
#
# s5 <- rbind(s5, pairs)
# s5 <- s5 %>% group_by(Time, fund_id) %>% transmute(
#   quartVol = sum(quartVol)
```

```r
# ) %>% as.data.frame %>% unique()
#
# remove(pairs)
# remove(dates)
# remove(fund_ids)
# remove(brokers)
# gc()
#
#
# s5 <- s5 %>%
#   group_by(fund_id) %>%
#   arrange(Time) %>%
#   mutate(
#     act4q = lag(quartVol, 4) + lag(quartVol, 3) + lag(quartVol, 2) + lag(quartVol, 1),
#     act2q = lag(quartVol, 2) + lag(quartVol, 1),
#     act8q = lag(quartVol, 4) + lag(quartVol, 3) + lag(quartVol, 2) + lag(quartVol, 1) +
#       lag(quartVol, 8) + lag(quartVol, 7) + lag(quartVol, 6) + lag(quartVol, 5)
#   )
#
# s5$act4q <- ifelse(s5$Time < as.POSIXct("2005-12-31"), 0, s5$act4q)
# s5$act8q <- ifelse(s5$Time < as.POSIXct("2006-12-31"), 0, s5$act8q)
# s5$act2q <- ifelse(s5$Time < as.POSIXct("2005-06-06"), 0, s5$act2q)
#
# #aggregate to count entrants and exits
# s5$entry <- ifelse(s5$quartVol != 0 & s5$act4q == 0, 1, 0)
# s5$exit <- ifelse(s5$act4q > 0 & s5$act2q == 0, 1, 0)
#
# ### New code starts here: between is just step 5 copied over
# #add indep var which is entry value lagged
# sub <- s5 %>%
#   group_by(fund_id) %>%
#   arrange(Time) %>%
#   mutate(
#     indep1 = ifelse(lag(entry,1) == 1, 1, 0),
#     indep2 = ifelse(lag(entry,2) == 1, 1, 0),
#     indep3 = ifelse(lag(entry,3) == 1, 1, 0),
#     indep4 = ifelse(lag(entry,4) == 1, 1, 0),
#     indep5 = ifelse(lag(entry,5) == 1, 1, 0),
#     indep6 = ifelse(lag(entry,6) == 1, 1, 0),
#     indep7 = ifelse(lag(entry,7) == 1, 1, 0),
#     indep8 = ifelse(lag(entry,8) == 1, 1, 0),
#     exit = ifelse(lag(exit) == 1 & exit == 1, 0, exit)
#   )

setwd("C:/Users/hyper/OneDrive/Documents/GitHub/Insurance-Corporate-Bonds")

sub <- read.csv("indepVarCalc.csv")

#Then need to rejoin data onto df for a by-transaction basis with quarterly data
df$Time <- as.POSIXct(df$Time)
df2 <- merge(df, sub, by = c('fund_id', 'Time'))

#With data set up, now want to run regressions
```

```
library(lfe)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
df2$Time <- as.factor(df2$Time)
df2$cusip <- as.factor(df2$cusip)
df2$indep <- df2$indep1
fm1 <- felm(epats ~ indep| Time * cusip | 0 | 0, data = df2)
df2$indep <- df2$indep2
fm2 <- felm(epats ~ indep| Time * cusip | 0 | 0, data = df2)
df2$indep <- df2$indep3
fm3 <- felm(epats ~ indep| Time * cusip | 0 | 0, data = df2)
df2$indep <- df2$indep4
fm4 <- felm(epats ~ indep| Time * cusip | 0 | 0, data = df2)
df2$indep <- df2$indep5
fm5 <- felm(epats ~ indep| Time * cusip | 0 | 0, data = df2)
df2$indep <- df2$indep6
fm6 <- felm(epats ~ indep| Time * cusip | 0 | 0, data = df2)
df2$indep <- df2$indep7
fm7 <- felm(epats ~ indep| Time * cusip | 0 | 0, data = df2)
df2$indep <- df2$indep8
fm8 <- felm(epats ~ indep| Time * cusip | 0 | 0, data = df2)


models <- list (
  fm1,
  fm2,
  fm3,
  fm4,
  fm5,
  fm6,
  fm7,
  fm8
)

results <- map(models, ~{
  model_summary <- summary(.x)
  data.frame(
    coef = model_summary$coefficients[, 1],
    s_error = model_summary$coefficients[, 2],
    t_stat = model_summary$coefficients[, 3],
    p_value = model_summary$coefficients[, 4],
    stringsAsFactors = FALSE
  )
}) %>%
```

```
  bind_rows(.id = "FixedEffectModel")

knitr::kable(results)
```

FixedEffectModel	coef	s_error	t_stat	p_value
1	0.0004224	0.0003415	1.236771	0.2161721
2	-0.0018014	0.0003325	-5.418085	0.0000001
3	-0.0023618	0.0003117	-7.576679	0.0000000
4	-0.0004836	0.0003189	-1.516306	0.1294421
5	-0.0004251	0.0003218	-1.320828	0.1865586
6	-0.0010165	0.0003127	-3.250769	0.0011509
7	-0.0017207	0.0002965	-5.803803	0.0000000
8	-0.0008368	0.0003009	-2.781334	0.0054136

```
m1 <- list(fm1, fm2, fm3)
m2 <- list(fm4, fm5, fm6)
m3 <- list(fm7, fm8)
```

```
stargazer(m1, title = "Results of regression with dummies, orig def. p1", type = 'latex', digits = NA, a
```

% Table created by stargazer v.5.2.3 by Marek Hlavac, Social Policy Institute. E-mail: marek.hlavac at gmail.com % Date and time: Thu, Mar 28, 2024 - 1:30:07 PM

Table 2: Results of regression with dummies, orig def. p1

	Dependent variable:		
	epats		
	(1)	(2)	(3)
indep	0.0004223558	-0.001801429^{***}	-0.002361752^{***}
	(0.0003414988)	(0.0003324844)	(0.0003117134)
Time FE	X	X	X
cusip FE	X	X	X
Observations	16,821,638	16,688,257	16,566,040
R^2	0.3761633	0.3761177	0.3760655
Adjusted R^2	0.3513035	0.3513489	0.351404
Residual Std. Error	0.08894521 (df = 16176987)	0.08928769 (df = 16051015)	0.08960464 (df = 15936151)
Note:			*p<0.1; **p<0.05; ***p<0.01

```
stargazer(m2, title = "Results of regression with dummies, orig def. p2", type = 'latex', digits = NA, a
```

% Table created by stargazer v.5.2.3 by Marek Hlavac, Social Policy Institute. E-mail: marek.hlavac at gmail.com % Date and time: Thu, Mar 28, 2024 - 1:30:43 PM

```
stargazer(m3, title = "Results of regression with dummies, orig def. p3", type = 'latex', digits = NA, a
```

% Table created by stargazer v.5.2.3 by Marek Hlavac, Social Policy Institute. E-mail: marek.hlavac at gmail.com % Date and time: Thu, Mar 28, 2024 - 1:30:49 PM

Table 3: Results of regression with dummies, orig def. p2

	Dependent variable:		
	epats		
	(1)	(2)	(3)
indep	−0.0004835512	−0.0004250821	−0.00101649***
	(0.0003189009)	(0.0003218299)	(0.0003126922)
Time FE	X	X	X
cusip FE	X	X	X
Observations	16,446,943	16,278,816	16,139,830
R^2	0.3759865	0.3759041	0.3758419
Adjusted R^2	0.3514004	0.3513563	0.3513727
Residual Std. Error	0.08991947 (df = 15823496)	0.09037972 (df = 15662746)	0.09076011 (df = 15530960)
Note:			*p<0.1; **p<0.05; ***p<0.01

Table 4: Results of regression with dummies, orig def. p3

	Dependent variable:	
	epats	
	(1)	(2)
indep	−0.001720668***	−0.0008368449***
	(0.0002964725)	(0.000300879)
Time FE	X	X
cusip FE	X	X
Observations	15,993,349	15,874,638
R^2	0.3757208	0.3756593
Adjusted R^2	0.3512953	0.3513183
Residual Std. Error	0.09117132 (df = 15391154)	0.09150372 (df = 15278961)
Note:		*p<0.1; **p<0.05; ***p<0.01