

IMPLIED COST OF CAPITAL WITH STATA

Dr. Jun Gu

E-mail: gujun@szu.edu.cn

Department of Accounting

School of Economics, Shenzhen University

本报告主要内容

- 内含资本成本 (Implied Cost of Capital) 的主要计算方法：
 - 内含资本成本简介
 - 内含资本成本计算流程
 - 内含资本成本计算的Stata实现
- Stata软件在数据监测(Data Inspection)方面的优势和作用：
 - 该部分论文中一般体现不出来；
 - 但对研究人员而言，这个工作可能比跑回归更花时间；
 - 以ICOC的基础数据为例介绍基础数据监测
- 给未来Stata版本功能升级的一些建议
 - 一家之言，仅供参考



内含资本成本简介

- 在会计和公司金融领域，我们经常需要知道企业的资本成本：

- 传统公司金融理论一般做法：

- 股权融资：假定其成本为 C_E ，其总额为 M_E
- 债务融资：假定其成本为 C_D ，其总额为 M_D
- 则该企业的综合资本成本为：

$$C_{Integrated} = \frac{M_E}{M_E + M_D} C_E + \frac{M_D}{M_E + M_D} C_D$$

- 但是此类做法会遇到一个问题：

- 我们必须使用事后变量来代理此类成本
 - 如通过财务费用来计算其利息成本，通过后续股票回报率计算风险等
- 若我们考虑企业的决策过程（Decision-making Process）
 - 企业则有可以是基于一个“预期”的资本成本来进行预算和控制
 - 上述的事后指标(Realized Index)可能就会有一些问题。



内含资本成本简介

- 从企业估值角度考虑，资本成本可以理解作为一种“内含报酬率”，即

$$P_t = \sum_{i=t}^{\infty} \frac{C_i}{(1+r)^i}$$

- 此种策略使用当前股价作为企业价值的替代变量
 - 即企业的价值等于企业未来现金流的折现值
 - 而其折现率则为企业的“内含资本成本”
- 但是这个模型的可行度并不高：
 - 未来现金流并不容易得到精确的“估计”
 - 我们更倾向于使用财报数据



内含资本成本简介

- 剩余收益模型（RIM）提供了一个重要思路（Ohlson, 1995）

$$E_{t+1} = E_t + NI_t - D_t = E_t + (1-k)NI_t$$

- 将该模型套入现金（股利）折现模型，我们可以得到：

$$\begin{aligned} P_t &= B_t + \sum_{i=1}^{\infty} \frac{E_t (NI_{t+i} - r_e B_{t+i-1})}{(1+r_e)^i} \\ &= B_t + \sum_{i=1}^{\infty} \frac{E_t [(ROE_{t+i} - r_e) B_{t+i-1}]}{(1+r_e)^i} \end{aligned}$$

- 因此，我们得到了一个基于会计数据的估值模型（Accounting-Based Valuation）



内含资本成本计算思路

- 基于RIM的估值模型给我们提供了一个思路（如GLS（2001））
 - P 为企业某一个时点的收盘价，同时作为企业的当前价值，该条件已知
 - 从 t 时点开始的未来盈余（Forecasted EPS），该条件已知
 - 这一步是以GLS为代表的方法精髓之所在；
 - 我们使用以分析师预测为基础的数据来替代未来盈余；
 - 但技术难点在于：分析师的预测是有限的
 - 此时，我们唯一未知的就是折现率- r
 - 这个 r 就是我们所感兴趣的内含报酬率



内含资本成本计算思路

■ 按照GLS的思路，未来预测的每股盈余（FEPS）会分为：

- 短期预测

■ 1-3年：分析师一般会提供3-5年预测；

■ 我们假定1-3年的业绩是线性增长，即 $FEPS_{t+1} = FEPS_t * (1 + LTG)$

■ 而长期增长率则默认为： $LTG = FEPS_{t+2} / FEPS_{t+1}$

- 长期预测

■ 我们假定企业未来的业绩会以一定的增长率持续增加，即

$$TV = \frac{FEPS_5 (1 + g)}{r_e (1 + r_e)^5}$$

■ 一般文献会使用GDP增长率来作为 g 的替代。



内含资本成本计算思路

■ 数据处理工作：

- 分析师数据处理：

■ 数据来源：CSMAR、CBSA，我们这里只讨论中国数据

■ 简单清洗：

- 盈余数据处理：

■ 1-3年的FEPS数据补齐；

■ 4-12年的FROE（注意不是FEPS）的计算，此处主要进行迭代

$$FROE_{t+1} = FROE_t + (ROE_{ind_median} - FROE_{t+3}) / 9$$

$$\begin{aligned} BV_{t+1} &= BV_t + (1-k)FEPS_{t+1} \\ &= BV_t + (1-k)FROE_{t+1} * BV_t \\ &= [1 + (1-k)FROE_{t+1}] * BV_t \end{aligned}$$



Stata简单实现流程

- 从CSMAR读取数据，我一般导出txt数据（方便合作者使用其他平台）
 - 导入过程本身很简单（注意：我们只使用年报预测数据）

```
import delimited AF_Forecast_`m'.txt, delimiter(tab) clear
ren rptdt anndats
ren fenddt fpendats

gen d1=trim(anndats)
gen d2=trim(fpendats)
drop anndats fpendats
```

Data Import

```
gen d1=trim(anndats)
gen d2=trim(fpendats)
drop anndats fpendats

forvalues i=1/2 {
    gen year_`i'=substr(d`i',1,4)
    gen month_`i'=substr(d`i',6,2)
    gen day_`i'=substr(d`i',9,2)
    gen d_`i'=year_`i'+month_`i'+day_`i'
    gen d_`i'_new=date(d_`i',"YMD")
    drop *_`i' d`i'
}
```

Data Conversion

国内分析师数据大致情况

Contains data from **feps_csamar.dta**

obs: 1,119,906

vars: 20

13 Aug 2018 16:06

size: 415,485,126

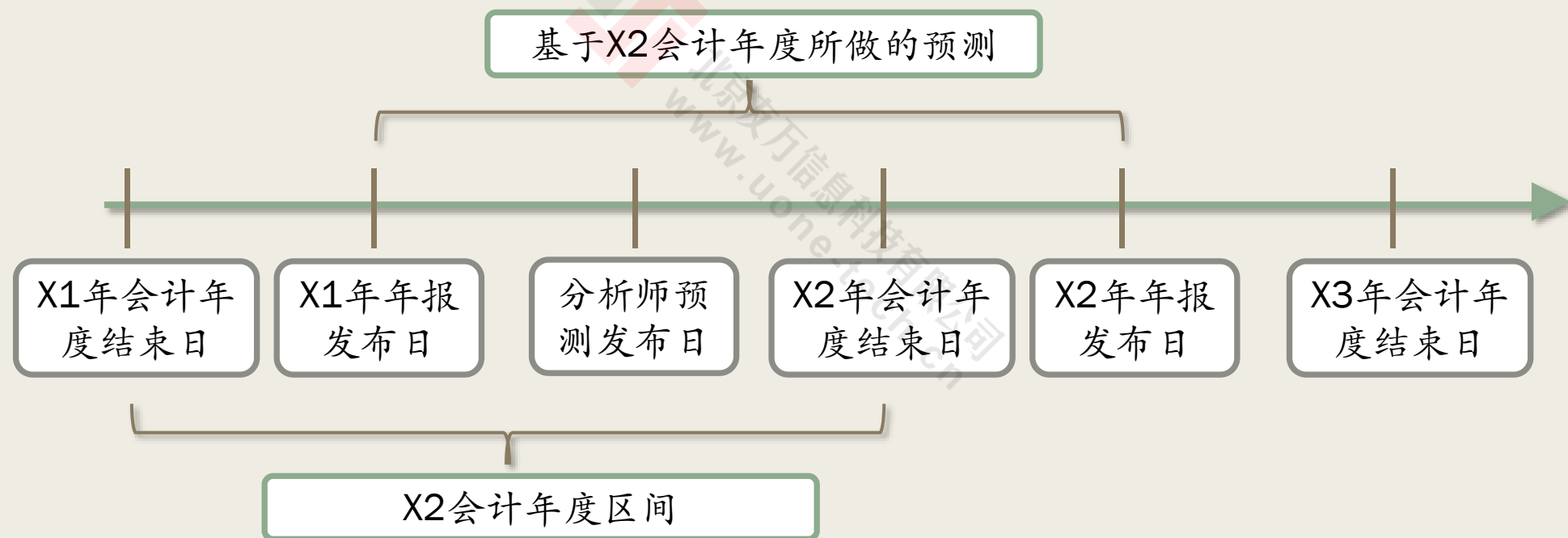
variable name	storage type	display format	value label
ananmid	str134	%134s	分析师ID
ananm	str73	%73s	分析师姓名
reportid	long	%10.0g	研究报告ID
institutionid	long	%10.0g	证券公司ID
brokern	str48	%48s	证券公司名称
feps	double	%10.0g	每股收益
fpe	double	%10.0g	市盈率
fnetpro	double	%10.0g	净利润
febit	double	%10.0g	息税前收入
febitda	double	%10.0g	扣除息、税、折旧及摊销前收入
fturnover	double	%10.0g	主营业务收入
fcfps	double	%10.0g	每股经营现金流
fbps	double	%10.0g	每股净资产
froa	double	%10.0g	总资产收益率
froe	double	%10.0g	净资产收益率
stkcd	long	%10.0g	证券代码
fpb	double	%10.0g	市净率
ftotalassetst~r	double	%10.0g	总资产周转率
anndats	float	%td	
fpendats	float	%td	

我们主要对这几个变量做文章!



如何来界定分析师所在的会计年度

- 事实上，会计年度与分析师所在年度界定存在一定程度的错配问题



- 因此，我们应该使用上年和本年的年报发布时间来作为其会计年度界定的上下界
- 这个动作错误会影响到我们后续其他财务数据的合并，导致“拉郎配”

如何来界定分析师所在的会计年度

■ 我们有两种不同思路来实现这个问题：

- 充分考虑国情的思路

- 我们的会计年度总是1月1日到12月31日，即与日历日期重合；
- 同时，GLS的算法只需要会计年度结束后第十个月的预测
- 因此，事实上，我们取得的分析师预测都是介于10月1日-10月31日
- 这个日期也完全没有与年报发布时间冲突的可能性

- 基于这个思路，我们直接使用分析师预测发布年份，即

- 若分析师的发布时间为2015年10月15日，则对应的会计年度为2015年12月31日

```
gen current_fiscal=year(anndats)
```

- 这种思路无法用于更复杂的跨国数据，或者其他更复杂的研究情形

- 下面我们介绍一种正式思路

如何来界定分析师所在的会计年度

■ 正式思路分为两步：

- 首先，取得各公司的本年与上年年报发布时间

```
gen pdate=l.reporting_date  
replace pdate=reporting_date-400 if pdate==.
```

- 其次，将分析师预测发布时间与上述数据合并。这里我用了一个土办法：

```
gen day_gap=reporting_date-pdate  
expand day_gap  
bys stkcd fiscal_end: replace anndats=pdate+_n  
sort stkcd anndats
```

- 最后，将上述数据与分析师数据进行合并

```
sort stkcd anndats  
Merge 1:1 stkcd anndats using fiscal_end_data, keepusing (fiscal_end)
```



分析师数据进行转置

- 按照前述模型要求，FEPS1-3将作为不同的变量使用，这就要求
 - 取得每一个公司分析师于不同年度对某一个特定会计期间预测的FEPS均值

```
bys stkcd fiscal_end fpendats: egen feps_mean=mean(feps)
keep stkcd fiscal_end fpi feps_mean
duplicates drop
```

- 然后，做一个新变量：gen fpi = year(fpendats)-year(fiscal_end)+1
 - 即fpi=1为当年预测；fpi=2为下一个会计年度预测，以此类推
- 最后转置：留个思考题吧~

	stkcd	fiscal_end	fpi	year	month	feps_mean
1	1	31dec2002	1	2002	6	.2
2	1	31dec2002	1	2002	7	.301
3	1	31dec2002	2	2003	4	.29
4	1	31dec2003	1	2003	6	.28
5	1	31dec2003	2	2004	3	.2
6	1	31dec2003	2	2004	4	.213
7	1	31dec2003	3	2004	4	.254

	stkcd	fiscal_end	year	month	feps_mean1	feps_mean2	feps_mean3
1	1	31dec2002	2002	6	.2	.	.
2	1	31dec2002	2002	7	.301	.	.
3	1	31dec2002	2003	4	.	.29	.
4	1	31dec2003	2003	6	.28	.	.
5	1	31dec2003	2004	3	.	.2	.
6	1	31dec2003	2004	4	.	.213	.254
7	1	31dec2004	2004	6	.217	.28	.

Stata简单实现流程

- 大多数情况下，1-3年的分析师预测并不全面，因此我们需要补足数据

```
263 //Filing Ups
264 gen ltg=.
265 forvalues i=1/4 {
266     local m=`i'+1
267     forvalues j=`m'/5 {
268         local k=`j'-'i'
269         replace ltg=(feps_mean`j'/feps_mean`i')^(1/`k')-1 if ltg==. & feps_mean`i'<. & feps_mean`i'!=0
270     }
271 }
272
273 replace ltg=0 if ltg<0
274 replace ltg=0.31 if ltg>0.31 //The median is around 31%
```

计算LTG

```
277 forvalues i=1/4 {
278     local j=`i'+1
279     forvalues k=`j'/5 {
280         replace feps_mean`i'=feps_mean`k'/(1+ltg)^(`k'-'i') if feps_mean`i'==. & feps_mean`k'<.
281     }
282 }
283
284 forvalue i=2/5 {
285     local j=`i'-1
286     forvalues k=`j'(-1)1 {
287         replace feps_mean`i'=feps_mean`k'*(1+ltg)^(`i'-'k') if feps_mean`i'==. & feps_mean`k'<.
288     }
289 }
```

补足EPS

Stata简单实现流程

- 下一步，我们准备1-12年的FEPS数据。其中：
 - 1-3年的FEPS直接以分析师预测作为基础，但BV依然需要迭代算出：

```
forvalues i=1/3 {  
    local j=`i'-1  
    gen froe`i'=feps_mean`i'/bv`j'  
    gen bv`i'=bv`j'+(1-dv_ratio)*feps_mean`i'  
}
```

- 4-12年则以行业中位ROE为基础进行差值 (Linear Interpolation)

```
gen gap=(froe3-roe_target)/9  
forvalues i=4/12 {  
    local j=`i'-1  
    gen froe`i'=froe`j'+gap  
    gen bv`i'=bv`j'+(1-dv_ratio)*froe`i'*bv`j'  
}
```



其他相关数据

- 按照RIM模型的要求，我们需要：
 - 股利支付率
 - 目前通用做法为该公司过去三年股利支付率的中位数；
 - 若上述指标为0或者其他异常值，可考虑使用该国过去三年所有公司同比率替代
 - 长期EPS的增长率
 - 注意，这里是Terminal Value部分的增长率，与LTG有着本质不同
 - 这里也是一个重要的“扑街”点
 - 目前做法主要使用所在国的GDP增长率代替
 - 意思是总能跟着大盘吧，但是梦想啊！
- 代码我就不给了，继续思考题吧~



计算思路

- 这里的主要策略是，使用Matlab来进行计算，但是需要解决：
 - 自动唤起Matlab；
 - 自动将Matlab的结果倒回到Stata中进行进一步操作；
 - 这个流程对于连续计算四种不同ICOC具有非常显著的意义。
- 由于Matlab的数据组织结构与Stata存在些许差异，我们需要进行一些调整：
 - 只把必要的数据放入Matlab；
 - 只把我们需要的结果传回Stata；
 - 同时，尽可能减少我的操作——懒。
- BTW：也有人用Excel的Goalseeker，这个真的太慢了



计算思路的Stata实现

■ 文件分割:

```
cd ../matlab
keep id stkcd fiscal_end
duplicates drop
sort id
save id, replace
```

	stkcd	fiscal_end	id
1	1	31dec2006	1
2	1	31dec2007	2
3	1	31dec2008	3
4	1	31dec2009	4
5	1	31dec2010	5
6	1	31dec2011	6
7	1	31dec2012	7
8	1	31dec2013	8
9	1	31dec2014	9
10	1	31dec2015	10

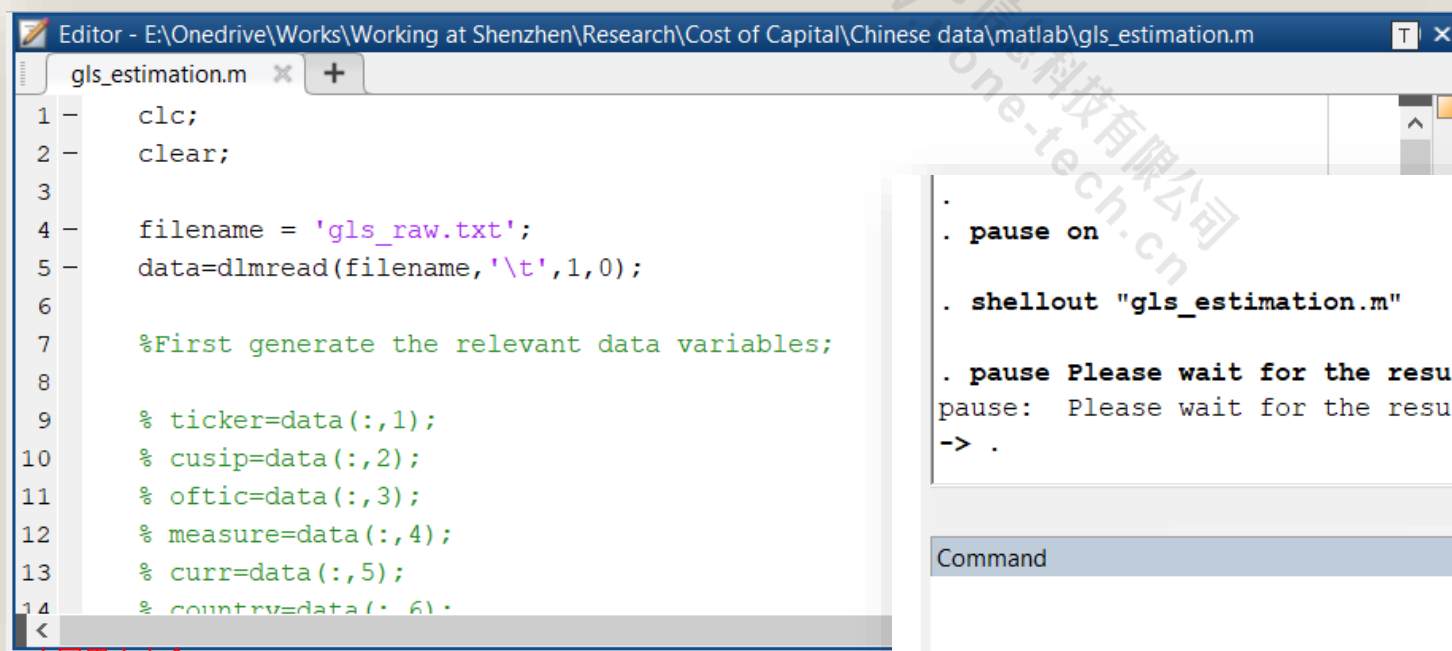
	id	mopnprc	inf	froe1	froe2	froe3	froe4
1	4448	11.5	.0065	.0444387	.0661929	.0838392	.0860118
2	4470	11	.0065	.1858026	.1610762	.2004159	.2136632
3	4437	9.21	.0065	.0734677	.0783295	.0832964	.0867893
4	469	5.44	.0005	.0557479	.1326244	.1533942	.1637006
5	940	8.38	.0005	.2051739	.2101723	.2144611	.2327737
6	986	5.85	.0005	.0274435	.0298315	.0323606	.0310612
7	3137	12.82	.0005	.367284	.3736818	.3210908	.3518018

计算思路的Stata实现

■ 唤起Matlab并暂停Stata

```
cd ..\matlab
export delimit using gls_raw.txt, delimiter(tab) replace

pause on
shellout "gls_estimation.m"
pause Please wait for the results from Matlab ...
pause off
```



The screenshot shows a Stata Editor window with the following code in the file `gls_estimation.m`:

```
1 - clc;
2 - clear;
3
4 - filename = 'gls_raw.txt';
5 - data=dlmread(filename,'\t',1,0);
6
7 %First generate the relevant data variables;
8
9 % ticker=data(:,1);
10 % cusip=data(:,2);
11 % oftic=data(:,3);
12 % measure=data(:,4);
13 % curr=data(:,5);
14 % country=data(:,6);
```

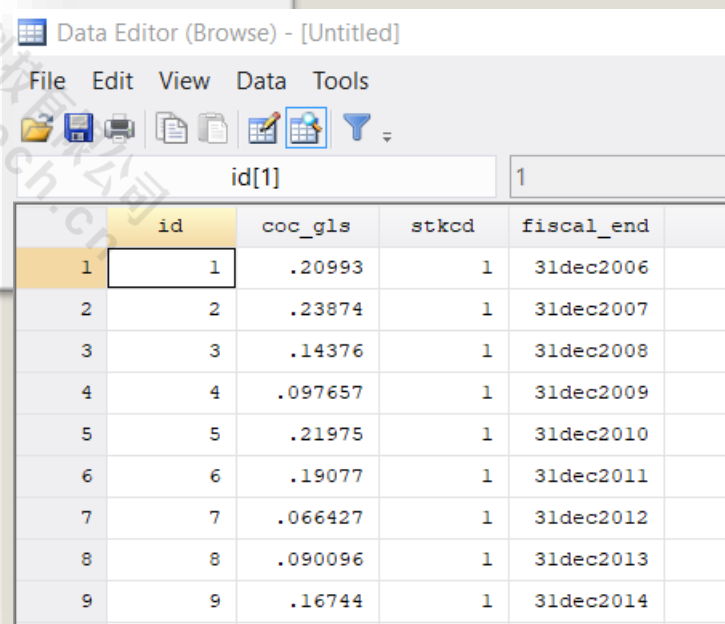
```
.
. pause on
. shellout "gls_estimation.m"
. pause Please wait for the results from Matlab ...
pause: Please wait for the results from Matlab ...
-> .
```

Command

计算思路的Stata实现

- 待Matlab执行结束，我们便可以将其结果传回Stata。但是这里要解决：
 - Matlab是直接以矩阵方式处理数据的，因此结果文件没有变量名；
 - Matlab -> TXT文档 -> Stata 这一流程

```
/*Merging Back GLS results*/  
import delimit using gls_result.txt, delimiter(tab) clear  
ren v1 id  
ren v2 coc_gls  
sort id  
  
merge 1:1 id using id  
keep if _merge==3  
drop _merge
```



Data Editor (Browse) - [Untitled]

File Edit View Data Tools

id[1] 1

	id	coc_gls	stkc	fiscal_end
1	1	.20993	1	31dec2006
2	2	.23874	1	31dec2007
3	3	.14376	1	31dec2008
4	4	.097657	1	31dec2009
5	5	.21975	1	31dec2010
6	6	.19077	1	31dec2011
7	7	.066427	1	31dec2012
8	8	.090096	1	31dec2013
9	9	.16744	1	31dec2014

实现思路总结

- 其实ICOC本身计算并不复杂，其主要工作在于：
 - 数据文件的顺序读入和顺序合并（import 和append）；
 - 数据的基础处理（限制月份、reshape等）
 - 数据的进一步处理（如1-3年的FEPS补足，4-12年的插值）
 - 与第三方软件的配合（我个人偏好使用txt作为媒介）
 - 第三方软件计算结果的重新导入
- 要点
 - Fully understand what you are doing;
 - 善用循环
- 至此，我们还遗留最后一个问题：数据本身会不会有什么问题？
 - ICOC结果对数据质量极其敏感



用Stata来进行数据监测 (Inspection)

- 其实数据监测能够帮我们发现一些数据的问题，其主要包括：
 - 数据供应商(Data Vendor)端可能存在的错误
 - 重复输入；
 - 手滑；
 - 莫名其妙的行为等；
 - 某些公司或者个人的匪夷所思行为：
 - 分析师刷存在感；
 - 数据库组织的习惯问题：
 - 全名与简称：比如APPL对Apple Inc. , J.Gu 对Jun Gu对Mr. J.Gu
 - 联名分析报告算一份还是算两份？（其实还有算三份的）
- 其实国内数据库组织简单，内容也比较“清淡”，所以问题不大
 - I/B E/S 的Detail File, DealScan等专治各种“不服”，大家了解下？



用Stata来进行数据监测 (Inspection)

- 我们回到ICOC计算问题。这里我们主要要解决的是分析师数据问题：
 - 相比较公司财务报表类数据，分析师数据似乎有点点“茫茫乱”
 - 我们手头有两个不同来源的分析师数据：
 - CSMAR：这个应该是大多数人的选择
 - CSBA：这个是我们因为某一个独立项目采购的
 - 于是，我们想知道分析师数据的质量到底如何：
 - 我们考虑两个维度：
 - 广度：谁涵盖了更多的分析师、证券公司；
 - 深度：谁包括了更多的项目；
 - 以及：他们是否可以相互验证？
 - 这个来源于我十年前读研时候学到的一个手段：CSMAR怼Wind



用Stata来进行数据监测 (Inspection)

- 首先，不同的机构会不会在同一天扎堆：
 - 左边是CSMAR，右边是CBSA：

```
bys stkcd brokern anndats fpendats: gen count=_N
```

```
tab count
```

count	Freq.	Percent	Cum.
1	1,119,324	99.95	99.95
2	582	0.05	100.00
Total	1,119,906		

count	Freq.	Percent	Cum.
1	1,134,485	99.61	99.61
2	4,396	0.39	100.00
3	24	0.00	100.00
Total	1,138,905	100.00	

用Stata来进行数据监测 (Inspection)

■ 我们来看下有没有重复行

ananm	reportid	institutio~d	brokern				sepe	
郑春明	8744415	10202	群益证券(香港)有限公司				1.68	
	8745823	10202	群益证券(香港)有限公司				1.68	
郑春明	8744415	10202	群益证券(香港)有限公司				1.81	
	8745823	10202	群益证券(香港)有限公司				1.81	
王逸,李薇	8746099	106067	北京高华证券有限责任公司				1.9	
王逸	8747095	106067	北京高华证券有限责任公司				1.9	
王逸	8747095	106067	北京高华证券有限责任公司				2.3	
章琪		东北证券股份有限公司	1.64	6.46	18694000	66658	有限责任公司	2.3
唐亚韞		东北证券股份有限公司	1.64	6.46	18694000	66658	有限责任公司	2.46
章琪		东北证券股份有限公司	1.9	5.58	21732000	78890	有限责任公司	2.46
唐亚韞		东北证券股份有限公司	1.9	5.58	21732000	78890	股份有限公司	1.901
章琪		东北证券股份有限公司	2.24	4.73	.	91907	股份有限公司	2.185
唐亚韞		东北证券股份有限公司	2.24	4.73	.	91907	股份有限公司	2.185
杨荣		中信建投证券股份有限公司	1.67	6.32	23833000	1.163e	股份有限公司	2.453
张玺		中信建投证券股份有限公司	1.67	6.32	23833000	1.163e	港)有限公司	1.56
Ning Ma/Jessica Wu/Nan Li		高盛高华证券有限责任公司	1.62	6	.		港)有限公司	1.56
李南		高盛高华证券有限责任公司	1.62	6	.			
潘嘉怡		东方证券股份有限公司	1.7	5.5	22865000	1.635e		
竺劲		东方证券股份有限公司	1.7	5.5	22865000	1.635e		
竺劲		东方证券股份有限公司	2.11	4.4	28395000	1.944e		
潘嘉怡		东方证券股份有限公司	2.11	4.4	28395000	1.944e		

一样的?



用Stata来进行数据监测 (Inspection)

- 同样的，我们想知道同一个机构或者分析师是否“刷榜”？

```
bys stkcd brokern ananm fpendats current_year: gen count=_N
```

- 这一步国内数据处理相对容易：
 - 我们的会计年度是严格限定为1月1日-12月31日，因此取年份即可

```
gen current_year = year(anndats)
```

- 但是这一步并不精确



	stkcd	a_code	star	brokern	b_code	feps_cbsa	fpe	fnetpro	foperev	anndats	fpendats	anndats_act	
157887	2321	A03801/A00860/A08881	0/0/0	申万宏源证券有限公司	B00132	.87	14	.		04dec2015	31dec2017	.	
157888	2321	A03801/A00860/A08881	0/0/0	申万宏源证券有限公司	B00132	.87	14	.		17dec2015	31dec2017	.	
157889	2321	A03801/A00860/A08881	0/0/0	申万宏源证券有限公司	B00132	.95	14	.		30dec2015	31dec2017	.	
157890	2321	A03801/A00860/A08881	0/0/0	申万宏源证券有限公司	B00132	.8	9	.		06jan2016	31dec2017	.	
157891	2321	A03801/A00860/A08881	0/0/0	申万宏源证券有限公司	B00132	.8	13	.		15jan2016	31dec2017	.	
157892	2321	A03801/A00860/A08881	0/0/0	申万宏源证券有限公司	B00132	.56	18	.		22jan2016	31dec2017	.	
157893	2321	A03801/A00860/A08881	0/0/0	申万宏源证券有限公司	B00132	.56	17	.		27jan2016	31dec2017	.	
157894	2321	A03801/A00860/A08881	0/0/0	申万宏源证券有限公司	B00132	.56	17	.		29jan2016	31dec2017	.	
157895	2321	A03801/A00860/A08881	0/0/0	申万宏源证券有限公司	B00132	.56	17	.		03feb2016	31dec2017	.	
157896	2321	A03801/A00860/A08881	0/0/0	申万宏源证券有限公司	B00132	.56	17	.		04feb2016	31dec2017	.	
157897	2321	A03801/A00860/A08881	0/0/0	申万宏源证券有限公司	B00132	.56	17	.		16feb2016	31dec2017	.	
157898	2321	A03801/A00860/A08881	0/0/0	申万宏源证券有限公司	B00132	.56	18	.		17feb2016	31dec2017	.	
157899	2321	A03801/A00860/A08881	0/0/0	申万宏源证券有限公司	B00132	.56	20	.		25feb2016	31dec2017	.	
157900	2321	A03801/A00860/A08881	0/0/0	申万宏源证券有限公司	B00132		19	.		01mar2016	31dec2017	.	
157901	2321	A03801/A00860/A08881	0/0/0	申万宏源证券有限公司	B00132	.56	20	.		03mar2016	31dec2017	.	
157902	2321	A03801/A00881	0/0	申万宏源证券有限公司	B00132	.56	21	.		04mar2016	31dec2017	.	
157903	2321	A03801/A00860/A08881	0/0/0	申万宏源证券有限公司	B00132	.56	21	.		07mar2016	31dec2017	.	
157904	2321	A03801	0	申万宏源证券有限公司	B00132	.56	20	.		10mar2016	31dec2017	.	
157905	2321	A03801/A00860/A08881	0/0/0	申万宏源证券有限公司	B00132	.56	20	.		11mar2016	31dec2017	.	
157906	2321	A03801/A00860/A08881	0/0/0	申万宏源证券有限公司	B00132	.56	18	.		18mar2016	31dec2017	.	
157907	2321	A03801/A00860/A08881		申万宏源证券有限公司	B00132	.56	21	.		22mar2016	31dec2017	.	
157908	2321	A03801/A00860/A08881		申万宏源证券有限公司	B00132	.56	21	.		23mar2016	31dec2017	.	
157909	2321	A03801/A00860/A08881		申万宏源证券有限公司	B00132	.56	19	.		25mar2016	31dec2017	.	
157910	2321	A03801/A00860/A08881		申万宏源证券有限公司	B00132	.56	22	.		28mar2016	31dec2017	.	
157911	2321	A03801/A00860/A08881		申万宏源证券有限公司	B00132	.56	21	.		30mar2016	31dec2017	.	
157912	2321	A03801/A00860/A08881		申万宏源证券有限公司	B00132	.56	20	.		31mar2016	31dec2017	.	
157913	2321	A03801/A00881		申万宏源证券有限公司	B00132	.56	20	.		01apr2016	31dec2017	.	
157914	2321	A03801/A00860/A08881	0/0/0	申万宏源证券有限公司	B00132	.56	20	.		04apr2016	31dec2017	.	
157915	2321	A03801/A00860/A08881	0/0/0	申万宏源证券有限公司	B00132	.56	20	.		05apr2016	31dec2017	.	
157916	2321	A03801/A00860/A08881	0/0/0	申万宏源证券有限公司	B00132	.71	16	388000	3235000	10apr2016	31dec2017	.	
157917	2321	A03801/A00860/A08881	0/0/0	申万宏源证券有限公司	B00132	.71	16	.		14apr2016	31dec2017	.	
157918	2321	A03801/A00860/A08881	0/0/0	申万宏源证券有限公司	B00132	.56	20	.	英	21apr2016	31dec2017	.	
157919	2321	A03801/A00860/A08881	0/0/0	申万宏源证券有限公司	B00132	.71	16	.		21apr2016	31dec2017	.	
157920	2321	A03801/A00860/A08881	0/0/0	申万宏源证券有限公司	B00132	.56	20	.		03may2016	31dec2017	.	
157921	2321	A03801/A00860/A08881	0/0/0	申万宏源证券有限公司	B00132	.71	14	.		09may2016	31dec2017	.	
157922	2321	A03801/A00860/A08881	0/0/0	申万宏源证券有限公司	B00132	.71	14	.		13may2016	31dec2017	.	
157923	2321	A03801/A00860/A08881	0/0/0	申万宏源证券有限公司	B00132	.71	13	.		17may2016	31dec2017	.	
157924	2321	A03801/A00860/A08881	0/0/0	申万宏源证券有限公司	B00132	.71	14	.		31may2016	31dec2017	.	
157925	2321	A03801/A00860/A08881	0/0/0	申万宏源证券有限公司	B00132	.71	14	.		02jun2016	31dec2017	.	
157926	2321	A03801/A00860/A08881	0/0/0	申万宏源证券有限公司	B00132	.71	14	.		08jun2016	31dec2017	.	

这叫打卡！！

用Stata来进行数据监测 (Inspection)

- 下面我们做一个重要工作。看看CSMAR和CBSA的重合度如何：
 - 如果重合度高：两者可以相互查漏补缺；
 - 如果重合度低：我信谁？
- 于是：

```
merge 1:1 stkcd anndats fpendats brokern ananm feps, keepusing (whatever you want)
```

- 结果。。。 (Master = CSMAR, Using = CBSA)

Result	# of obs.	
not matched	1,375,235	
from master	704,521	(_merge==1)
from using	670,714	(_merge==2)
matched	434,384	(_merge==3)



用Stata来进行数据监测 (Inspection)

- 考虑到部分分析报告有多位分析师参与，我想这方面是不是有问题：

- 两个dataset所用的分隔符不同，而且万一人数对不上呢？
- 于是，我决定采用国内通行学术标准——只认第一作者！

```
split ananm, p("/", ",")
```

- 我们对前述命令稍加修改

```
merge 1:1 stkcd anndats fpendats brokern first_author feps, keepusing (whatever you want)
```

- 我们得到

Result	# of obs.	
not matched	828,813	
from master	431,310	(_merge==1)
from using	397,503	(_merge==2)
matched	707,595	(_merge==3)

用Stata来进行数据监测 (Inspection)

- 上述结果使得选择颇为困难，于是我打算换一个思路：
 - 这俩数据库所涵盖的分析师和机构范围是否一致？

```
use datasetname, clear  
keep brokern first_author  
duplicates drop  
sort brokern first_author  
save dataset_namelist, replace
```

- 结果如下：

```
. merge 1:1 brokern first_author using csmar_analyst_list, keepusing(source)  
(note: variable source was str4, now str5 to accommodate using data's values)  
(note: variable first_author was str16, now str18 to accommodate using data's values)
```

Result	# of obs.	
not matched	3,905	
from master	1,845	(_merge==1)
from using	2,060	(_merge==2)
matched	5,845	(_merge==3)



用Stata来进行数据监测 (Inspection)

- 我们进一步来直接比较两者所包含的金融机构：

- Master = CBSA, Using = CSMAR

Result	# of obs.	
not matched	57	
from master	20	(_merge==1)
from using	37	(_merge==2)
matched	164	(_merge==3)

- 你们选哪个？



数据监测后的方案调整

- 根据数据监测的结果，我们发现，前述数据处理方案似乎有点小瑕疵；
- 所以我们进行一些必要的调整：
 - 考虑到其优势，我们使用CSMAR作为主要数据：
 - 更好的覆盖率；
 - 相对更少的“刷榜”类数据
 - 对于同一个机构和同一个分析师所发布的相同业绩预测而言：
 - 我们仅保留其最早的一条记录，同时我们“认定”第一作者优先；
 - 直接踢掉“恶意刷榜”者所有的记录——对，我就这么任性
 - 考虑补充CSMAR未包括而CBSA包括的分析师预测发布信息：
 - 这部分有一定的风险；
 - 因此我们目前测试方案很保守，仅仅包括CSMAR未包括的分析师和机构的信息
 - 我们不纳入CSMAR已经包括的机构和分析师，但时间和结果对不上的CBSA记录



关于Stata软件的体会

■ Stata的优点在于：

- 提供了单语句交互模式的工作方式（类似于DOS和Linux）；
- 语法规则松散，也不需要声明变量类型，更不用编译；
- 大量第三方的Package支持（善用ssc install和findit）

■ 可考虑未来升级的部分：

- 考虑完整的SQL支持；
- 提供更加完善的编辑器功能；
 - 如最常见的单步调试功能
- 如有可能，提供一个调试窗口，使得我们更容易看到变量值的变化



A Small Final Trick

- Don't stay up too late, otherwise, something can happen 亮点自寻

```
110
111 merge 1:1 stكد ananml brokern anndats fpendats using feps_csmar_splited, keep
112 tab _merge
113
114 preserve
115     keep if _merge==1
116     save cbsa_unmatched_splited, replace
117     sort stكد ananml brokern feps fpendats anndats
118     ren ananml first_author
119     drop ananm*
120     restore
121
122 preserve
123     keep if _merge==2
124     save csmar_unmatched_splited, replace
125     sort stكد ananml brokern feps fpendats anndats
126     ren ananml first_author
127     drop ananm*
128     gen anndats_csmar=anndats
129     format anndats_csmar %td
130     restore
131
```

```
. use cbsa_unmatched_splited, clear

. merge 1:1 stكد first_author brokern feps fpendats, keepusing (anndats_csmar)
using required
r(100);

end of do-file

r(100);
```

