

```

1 *****;
2 *           Program for initial data collection           *;
3 *           Last updated: 29 August 2018                 *;
4 *****;
5
6 ** Preamble;
7
8 options user="YOUR_DIR";
9 libname oldwork "YOUR_DIR";
10 libname sas "YOUR_DIR";
11 libname wrds "YOUR_DIR";
12 libname taq 'YOUR_DIR';
13 libname raven 'YOUR_DIR';
14 libname ww "YOUR_DIR";
15 libname zacks "YOUR_DIR";
16
17 *****;
18 ***** Basic sample selection (assets, returns, remove REITs) *****;
19 *****;
20
21 *1. Download CRSP/Compustat data;
22 %macro basic_info;
23
24 * pull data from compustat;
25     proc sql;
26         create      table compq as
27         select      a.gvkey, a.datadate, a.rdq, (a.fyearq * 10) + a.fqtr
28 as fyearqtr, a.tic, a.epsfxq,
29                 a.atq, a.prccq
30         from wrds.compfundq a
31         where ((indfmt = 'INDL') and (consol = 'C') and (popsrc = 'D')
32 and (datafmt = 'STD'))
33               and '01jan2012'd le rdq le '10nov2015'd
34         order by gvkey, fyearqtr, atq descending, epsfxq descending,
35 prccq descending;
36     quit;
37
38
39     * eliminate duplicate quarters - very small number so just ignore them;
40     proc sort data= compq nodupkey;
41         by gvkey fyearqtr;
42     run;
43
44 *Merge in CRSP data. This follows the CRSP/Compustat Merge procedures as suggested
45 by WRDS and extended link (Richard Price);
46 proc sql;
47 create table compq2 as
48 select a.*, b.lpermno as permno, b.linkdt, b.extlinkenddt
49 from compq as a left join wrds.mycstlink as b
50 on a.gvkey=b.gvkey
51 where b.linkdt <= a.rdq <= b.extlinkenddt;
52
53 select * from _last_;
54
55 data compq2;
56     set compq2;
57     if missing (permno) then
58         delete;
59 run;
60
61 data stocknames;

```

```

62 set wrds.stocknames_new (keep=namedt nameenddt permno ticker comnam hexcd exchcd
63 shrcd shrcls);
64 if not(missing(namedt));
65 run;
66
67 PROC SQL;
68     CREATE TABLE compq3 AS
69     SELECT a.ticker, a.comnam, a.hexcd, a.exchcd, a.shrcd, a.shrcls, b.*
70     FROM stocknames as a RIGHT JOIN compq2 as b
71     ON a.permno = b.permno
72        AND (a.namedt <= b.rdq OR a.namedt = .B)
73        AND (b.rdq <= a.nameenddt OR a.nameenddt>= '30jun2017'd); *this is
74 because the end date is 6/30/17 in the stocknames dataset;
75 QUIT;
76
77
78 proc sort data=compq3 nodupkey;
79     by gvkey fyearqtr; run;
80
81 * require total assets and quarter end share price from Compustat;
82 ** oldwork.EA_ticker3_hand has the dataset of matched AP articles;
83
84 data startdset (keep = gvkey datadate rdq fyearqtr permno sumAP an_article AI_cov
85 AP_cov post pre);
86     set oldwork.EA_ticker3_hand;
87 run;
88
89 ** require basic assets and price data;
90 proc sql; create table compq3 as select a.*, b.atq, b.prccq, b.exchcd, b.shrcd,
91 b.comnam, b.ticker, b.tic, b.epsfxq
92 from startdset as a left join compq3 as b
93 on a.gvkey=b.gvkey and a.fyearqtr=b.fyearqtr; quit;
94
95 data compq3;
96 set compq3;
97 if atq>0 and prccq>0;
98 if ticker='' then ticker=tic;
99 drop tic;
100 run;
101
102 proc sort data= compq3  nodupkey;
103     by gvkey fyearqtr;
104     run;
105 * take out trusts, closed-end funds, REITs;
106
107 data compq4;
108 set compq3;
109 if (shrcd=10 or shrcd=11 or shrcd=12 or shrcd=30 or shrcd=31 or shrcd=32);
110 run;
111
112 proc sort data= compq4  nodupkey;
113     by gvkey fyearqtr;
114     run;
115
116 * require [0,2] stock returns;
117 proc sql;
118 create table compq4
119 as select a.*, b.trading_date, b.trading_date_plus1, b.trading_date_plus2
120 from compq4 as a left join sas.trading_dates_final as b
121 on a.rdq=b.orig_date;
122 quit;

```

```

123
124 proc sql;
125 create table compq4
126 as select a.*, b.ret as ret_0
127 from compq4 as a left join wrds.crspdsf as b
128 on a.permno=b.permno and intck('day',a.trading_date,b.date)=0;
129 quit;
130
131 proc sql;
132 create table compq4
133 as select a.*, b.ret as ret_1
134 from compq4 as a left join wrds.crspdsf as b
135 on a.permno=b.permno and intck('day',a.trading_date_plus1,b.date)=0;
136 quit;
137
138 proc sql;
139 create table compq5
140 as select a.*, b.ret as ret_2
141 from compq4 as a left join wrds.crspdsf as b
142 on a.permno=b.permno and intck('day',a.trading_date_plus2,b.date)=0;
143 quit;
144
145 data compq6;
146 set compq5;
147 ret_0_2=(1+ret_0)*(1+ret_1)*(1+ret_2)-1;
148 if ret_0_2^='';
149 run;
150
151 proc sort data=compq6 nodupkey out=crsp;
152 by gvkey fyearqtr; run;
153
154 %mend basic_info;
155
156 *****;
157 ***** Zacks and IBES info *****;
158 *****;
159
160 %macro zacks_ibes;
161
162 * match in Zacks data;
163 data stocknames;
164 set wrds.stocknames_new (keep=namedt nameenddt permno cusip ncusip ticker comnam
165 hexcd exchcd shrccd shrcls);
166 if not(missing(namedt));
167 run;
168
169 * get current cusip for each obs;
170
171 PROC SQL;
172 CREATE TABLE cusips AS
173 SELECT a.cusip, b.*
174 FROM stocknames as a RIGHT JOIN crsp as b
175 ON a.permno = b.permno
176 AND (a.namedt <= b.rdq OR a.namedt = .B)
177 AND (b.rdq <= a.nameenddt OR a.nameenddt >= '30jun2017'd); *this is
178 because the end date is 6/30/15 in the stocknames dataset;
179 QUIT;
180
181 proc sort data=cusips nodupkey;
182 by gvkey fyearqtr; run;
183

```

```

184 * get current cusip for each Zacks id;
185 PROC SQL;
186     CREATE TABLE zacks_company_info AS
187     SELECT substr(a.cusip,1,8) as cusip, b.*
188     FROM stocknames as a RIGHT JOIN zacks.company_info as b
189     ON substr(a.ncusip,1,8) = substr(b.cusip,1,8);
190     QUIT;
191
192 * for multiple zacks IDs, take the current one;
193 proc sort data=zacks_company_info nodup;
194 by cusip current_inactive_code; run;
195
196 proc sort data=zacks_company_info nodupkey;
197 by cusip; run;
198
199 * match Zacks id;
200 * get Zacks consensus, actual, report date, number of analysts;
201 proc sql; create table zacks
202 as select a.*, b.zid
203 from cusips as a left join zacks_company_info as b
204 on substr(a.cusip,1,8)=substr(b.cusip,1,8); quit;
205
206 proc sort data=zacks nodupkey;
207 by gvkey fyearqtr; run;
208
209 proc sql; create table zacks1
210 as select a.*, b.actual_eps, b.consensus_eps, b.number_of_est, b.report_date
211 from zacks as a left join zacks.eps_surp as b
212 on month(b.report_date)=month(a.rdq) and year(b.report_date)=year(a.rdq) and
213 a.zid=b.zid;
214 quit;
215
216 proc sort data=zacks1 nodupkey;
217 by gvkey fyearqtr; run;
218
219 * get share and price correction factor, price from CRSP;
220 proc sql; create table zacks1
221 as select a.*, b.trading_date as datadate_trading
222 from zacks1 as a left join sas.trading_dates_final as b
223 on a.datadate=b.orig_date; quit;
224
225 proc sql;
226     create table zacks1
227     as select a.*, b.cfacsadj as crspshradj_qtr_end, b.cfacspr as
228     crspprcadj_qtr_end, abs(b.prc) as crsp_price, b.shrout
229     from zacks1 as a left join wrds.crspdsf as b
230     on a.permno=b.permno and a.datadate_trading=b.date;quit;
231
232 data zacks1;
233 set zacks1;
234 if crspshradj_qtr_end='' then crspshradj_qtr_end=1;
235 if crspprcadj_qtr_end='' then crspprcadj_qtr_end=1;
236 run;
237
238 ***PREP WORK to get IBES report date, consensus, actual, number of estimates;
239
240 *Pulling IBES info and joining actual and forecast;
241 *Select criteria for actual EPS;
242 %let actual_screen=(2009<=year(pends)<=2016 and usfirm=1 and measure='EPS' and
243 pdicity='QTR' and value ne .);
244 %put &actual_screen.;

```

```

245         *Select criteria for forecasts;
246 %let forecast_screen= (2009<=year(fpedats)<=2016 and usfirm=1 and measure='EPS' and
247 estflag='P' and fiscalp='QTR');
248
249         *Pulling actual EPS from Unadjusted Detail History Actuals File;
250 data actual;
251     set wrds.ibes_actu_epsus(keep=ticker cusip pends anndats measure pdicity
252 value usfirm);
253                                     *pendis is period end date,anndats is announcement/report
254 date;
255     where &actual_screen;run;
256
257         *Pulling analyst forecast info from summary unadjusted forecast table;
258 data forecast;
259     set wrds.ibes_statsumu_epsus (keep=ticker cusip oftic cname statpers measure
260 fiscalp fpi numest medest meanest
261     stdev usfirm fpedats estflag); *fpedats is period end date,
262 statpers is forecast date;
263     where &forecast_screen;
264     drop measure usfirm;run;
265
266         *Pulling any forecasts within 1 year before fiscal period end and through
267 report date;
268 proc sql;
269     create table ibes
270     as select a.*, b.*
271     from (actual as a join forecast as b
272     on a.ticker=b.ticker and a.pends=b.fpedats and a.pdicity=b.fiscalp
273     and intnx('month',a.pends,-
274 12,'beginning')<=b.statpers<=a.anndats)
275     order by a.ticker, a.pends, a.pdicity, b.statpers desc;quit;
276
277 *****;
278
279         *Matching IBES tickers to permnos;
280 proc sql;
281     create table ibes3
282     as select a.*, b.permno, b.score
283     from ibes as a left join wrds.iclink (keep=ticker permno score where=(score
284 in(0,1,2))) as b
285     on a.ticker=b.ticker
286     order by a.ticker, b.permno, a.fpedats, a.statpers, b.score;quit;
287
288         **Getting additional info and calculating earnings surprise from IBES data;
289 proc sql;
290     create table ibes4
291     as select a.*, b.trading_date as ibes_anndats_trading
292     from ibes3 as a left join sas.trading_dates_final as b
293     on a.anndats=b.orig_date;quit;
294
295 proc sql;
296     create table ibes5
297     as select a.*, b.trading_date as ibes_forecast_trading
298     from ibes4 as a left join sas.trading_dates_final as b
299     on a.statpers=b.orig_date;quit;
300
301         *Pulling CRSP adjustment factors for each IBES date and estimating earnings
302 surprise variable;
303         *Pulling CRSP conversion factor for report dates;
304 proc sql;
305     create table ibes6

```

```

306         as select distinct a.*, b.cfacschr as crspshradj_ibes_report, b.cfacpr as
307 crspprcadj_ibes_report
308         from ibes5 as a left join wrds.crspdsf as b
309             on a.permno=b.permno and a.ibes_anndats_trading=b.date;quit;
310         *Pulling CRSP conversion factor for forecast dates;
311     proc sql;
312         create table ibes7
313         as select distinct a.*, b.cfacschr as crspshradj_ibes_forecast, b.cfacpr as
314 crspprcadj_ibes_forecast
315         from ibes6 as a left join wrds.crspdsf as b
316             on a.permno=b.permno and a.ibes_forecast_trading=b.date;quit;
317
318         *Then creating earnings surprise variable and capturing IBES announcement
319 date;
320
321     proc sql;
322         create table zacks2
323         as select distinct a.*,b.*,
324             (b.value - ((b.medest/b.crspshradj_ibes_forecast)*b.crspshradj_ibes_report))
325 as ibes_ue_raw,
326             (b.value -
327             ((b.medest/b.crspshradj_ibes_forecast)*b.crspshradj_ibes_report))
328             /((a.crsp_price/a.crspprcadj_qtr_end)*b.crspprcadj_ibes_report)
329 as ibes_ue
330         from zacks1 as a left join ibes7 as b
331             on a.permno=b.permno and b.permno ne . and
332 year(a.datadate)=year(b.fpedats) and month(a.datadate)=month(b.fpedats)
333             and a.rdq=b.anndats;quit;
334
335
336     **Keeping only most recent forecast period and Dropping duplicate matches to IBES
337 tickers based on ..
338         1) smallest (best) ICLINK score
339         2) (among score ties) smallest difference between IBES actual amount
340 and Compustat actual amount;
341     data zacks2;
342         set zacks2;
343         temp_ibes_diff = abs(value - epsfxq);run;
344     proc sort data=zacks2; by gvkey fyearqtr descending statpers score
345 temp_ibes_diff;run;
346     proc sort data=zacks2 nodupkey; by gvkey fyearqtr;run;
347
348     * import WSH data;
349     proc import datafile="YOUR_DIR/WSH_EA_History_2006_2016.xlsx" out=wsh2012 dbms=xlsx
350 replace;
351 SHEET="2012";
352 getnames=yes;
353 run;
354
355     proc import datafile="YOUR_DIR/WSH_EA_History_2006_2016.xlsx" out=wsh2013 dbms=xlsx
356 replace;
357 SHEET="2013";
358 getnames=yes;
359 run;
360
361     proc import datafile="YOUR_DIR/WSH_EA_History_2006_2016.xlsx" out=wsh2014 dbms=xlsx
362 replace;
363 SHEET="2014-2";
364 getnames=yes;
365 run;
366

```

```

367 proc import datafile="YOUR_DIR/WSH_EA_History_2006_2016.xlsx" out=wsh2015 dbms=xlsx
368 replace;
369 SHEET="2015";
370 getnames=yes;
371 run;
372
373 proc import datafile="YOUR_DIR/WSH_EA_History_2006_2016.xlsx" out=wsh2016 dbms=xlsx
374 replace;
375 SHEET="2016";
376 getnames=yes;
377 run;
378
379 data wsh1;
380 set wsh2012 (rename=(ticker_symbol=ticker Company_Name=CompanyName
381 Quarter_End_Date=QuarterEnd Fiscal_Year=FiscalYear)) wsh2013
382 (rename=(ticker_symbol=ticker Company_Name=CompanyName
383 Quarter_End_Date=QuarterEnd))
384 wsh2014 (rename=(ticker_symbol=ticker Company_Name=CompanyName
385 Quarter_End_Date=QuarterEnd Fiscal_Year=FiscalYear) drop=EPS_OC EPS_OC_CUR);
386 if Fiscal_Year='2013?' then FiscalYear=2013;
387 if FiscalYear='' then FiscalYear=Fiscal_Year;
388 run;
389
390 data wsh2;
391 set wsh2015 (drop=EPS_OC EPS_OC_CUR) wsh2016 (drop=EPS_OC EPS_OC_CUR);
392 run;
393
394 data wsh;
395 set wsh1 (drop=Fiscal_Year) wsh2;
396 fyearqtr1 = cats(trim(FiscalYear), trim(left(substr(Quarter, index(Quarter, 'Q') +
397 1)))));
398 EA_date=datepart(EA_DateTime);
399 if (index(ticker, '.A')>0 or index(ticker, '.B')>0) then ticker=scan(ticker,1, '.');
400 *takes care of dual class, such as BRK.A and BRK.B;
401 if (index(ticker, ':')=0 and index(ticker, '.')=0);
402 fyearqtr=input(fyearqtr1,8.);
403 drop fyearqtr1;
404 run;
405
406 * match into zacks2;
407 proc sql; create table zacks2 as select distinct a.*, b.EPS as EPS_WSH,
408 b.CompanyName as CompanyName_WSH, b.EA_date as EA_date_WSH
409 from zacks2 as a left join wsh as b
410 on a.ticker=b.ticker and a.datadate=b.QuarterEnd; quit;
411
412 proc sort data=zacks2 nodup;
413 by gvkey fyearqtr descending EPS_WSH;
414 run;
415
416 proc sort data=zacks2 nodupkey;
417 by gvkey fyearqtr;
418 run;
419
420 * clean up WSH company name and see if close to crsp comnam;
421 data zacks2;
422 set zacks2;
423 company_wsh=compbl(upcase(compress(CompanyName_WSH, '.'))) || '/';
424 company_wsh=transtrn(company_wsh, 'COMPANY INC /', trimn(''));
425 company_wsh=transtrn(company_wsh, 'CORP NEW /', trimn(''));
426 company_wsh=transtrn(company_wsh, 'CORP DEL /', trimn(''));
427 company_wsh=transtrn(company_wsh, 'CO DEL /', trimn(''));

```

```

428 company_wsh=transtrn(company_wsh,'DEL NEW /',trimn(''));
429 company_wsh=transtrn(company_wsh,'INC NEW /',trimn(''));
430 company_wsh=transtrn(company_wsh,'CO INC /',trimn(''));
431 company_wsh=transtrn(company_wsh,'INC /',trimn(''));
432 company_wsh=transtrn(company_wsh,'CORP /',trimn(''));
433 company_wsh=transtrn(company_wsh,'PLC /',trimn(''));
434 company_wsh=transtrn(company_wsh,'P L C /',trimn(''));
435 company_wsh=transtrn(company_wsh,'LTD /',trimn(''));
436 company_wsh=transtrn(company_wsh,'COR /',trimn(''));
437 company_wsh=transtrn(company_wsh,'CO /',trimn(''));
438 company_wsh=transtrn(company_wsh,'IN /',trimn(''));
439 company_wsh=transtrn(company_wsh,'CORP II /',trimn(''));
440 company_wsh=transtrn(company_wsh,'GRP /',trimn(''));
441 company_wsh=transtrn(company_wsh,'GROUP /',trimn(''));
442 company_wsh=transtrn(company_wsh,'GRP LT /',trimn(''));
443 company_wsh=transtrn(company_wsh,'NEW /',trimn(''));
444 run;
445
446 data zacks2;
447 set zacks2;
448 wsh_dist=compged(lowercase(company_wsh),lowercase(company));
449 run;
450
451 * require non-missing Zacks, IBES, or WSH EA date;
452 data zacksoribes;
453 set zacks2;
454 if (anndats^='' or report_date^='' or EA_date_WSH^='');
455 format EA_date_WSH date9.;
456 run;
457
458 proc sort data=zacksoribes nodupkey;
459 by gvkey fyearqtr; run;
460
461 * require one of Zacks EA , IBES EA, or WSH EA to be same as rdq;
462 data same_ea;
463 set zacksoribes;
464 if (anndats=rdq or report_date=rdq or EA_date_WSH=rdq);
465 run;
466
467
468 proc sort data=same_ea nodupkey out=zacks_rdq;
469 by gvkey fyearqtr; run;
470
471 %mend zacks_ibes;
472
473 *****;
474 ***** Correct rdq and EA announcement times *****;
475 *****;
476
477 %macro anntimes;
478
479 ** pull ibes announce times;
480
481 data anntimes;
482 set zacks_rdq (keep = gvkey fyearqtr permno datadate rdq epsfxq);
483 run;
484
485 *Pulling IBES info and joining actual and forecast;
486 *Select criteria for actual EPS;
487 %let actual_screen=(2009<=year(pends)<=2016 and usfirm=1 and measure='EPS' and
488 pdicity='QTR' and value ne .);

```



```

489      *Select criteria for forecasts;
490 %let forecast_screen= (2009<=year(fpedats)<=2016 and usfirm=1 and measure='EPS' and
491 estflag='P' and fiscalp='QTR');
492
493      *Pulling actual EPS from Unadjusted Detail History Actuals File;
494 data actual;
495     set wrds.ibes_actu_epsus(keep=ticker cusip pends anndats anntims measure
496 pdicity value usfirm);
497     *pendis is period end date,anndats is announcement/report
498 date;
499     where &actual_screen;run;
500
501      *Pulling analyst forecast info from summary unadjusted forecast table;
502 data forecast;
503     set wrds.ibes_statsumu_epsus (keep=ticker cusip oftic cname statpers measure
504 fiscalp fpi numest medest meanest
505 stdev usfirm fpedats estflag); *fpedats is period end date,
506 statpers is forecast date;
507     where &forecast_screen;
508     drop measure usfirm;run;
509
510      *Pulling any forecasts within 1 year before fiscal period end and through
511 report date;
512 proc sql;
513     create table ibes
514     as select a.*, b.*
515     from (actual as a join forecast as b
516         on a.ticker=b.ticker and a.pends=b.fpedats and a.pdicity=b.fiscalp
517         and intnx('month',a.pends,-
518 12,'beginning')<=b.statpers<=a.anndats)
519     order by a.ticker, a.pends, a.pdicity, b.statpers desc;quit;
520
521 *****;
522
523      *Matching IBES tickers to permnos;
524 proc sql;
525     create table ibes3
526     as select a.*, b.permno, b.score
527     from ibes as a left join wrds.iclink (keep=ticker permno score where=(score
528 in(0,1,2))) as b
529     on a.ticker=b.ticker
530     order by a.ticker, b.permno, a.fpedats, a.statpers, b.score;quit;
531
532      **Getting additional info and calculating earnings surprise from IBES data;
533 proc sql;
534     create table ibes4
535     as select a.*, b.trading_date as ibes_anndats_trading
536     from ibes3 as a left join sas.trading_dates_final as b
537     on a.anndats=b.orig_date;quit;
538
539 proc sql;
540     create table ibes5
541     as select a.*, b.trading_date as ibes_forecast_trading
542     from ibes4 as a left join sas.trading_dates_final as b
543     on a.statpers=b.orig_date;quit;
544
545      *Pulling CRSP adjustment factors for each IBES date and estimating earnings
546 surprise variable;
547      *Pulling CRSP conversion factor for report dates;
548 proc sql;
549     create table ibes6

```

```

550         as select distinct a.*, b.cfacschr as crspshradj_ibes_report, b.cfacpr as
551 crspprcadj_ibes_report
552         from ibes5 as a left join wrds.crspdsf as b
553             on a.permno=b.permno and a.ibes_anndats_trading=b.date;quit;
554         *Pulling CRSP conversion factor for forecast dates;
555     proc sql;
556         create table ibes7
557         as select distinct a.*, b.cfacschr as crspshradj_ibes_forecast, b.cfacpr as
558 crspprcadj_ibes_forecast
559         from ibes6 as a left join wrds.crspdsf as b
560             on a.permno=b.permno and a.ibes_forecast_trading=b.date;quit;
561
562         *Then creating earnings surprise variable and capturing IBES announcement
563 date;
564
565     proc sql;
566         create table anntimes
567         as select distinct a.*,b.*
568             from anntimes as a left join ibes7 as b
569                 on a.permno=b.permno and b.permno ne . and
570 year(a.datadate)=year(b.fpedats) and month(a.datadate)=month(b.fpedats)
571                 and a.rdq=b.anndats;quit;
572
573     **Keeping only most recent forecast period and Dropping duplicate matches to IBES
574 tickers based on ..
575         1) smallest (best) ICLINK score
576         2) (among score ties) smallest difference between IBES actual amount
577 and Compustat actual amount;
578     data anntimes;
579         set anntimes;
580         temp_ibes_diff = abs(value - epsfxq);run;
581     proc sort data=anntimes; by gvkey fyearqtr descending statpers score
582 temp_ibes_diff;run;
583     proc sort data=anntimes nodupkey; by gvkey fyearqtr;run;
584
585     ** pull Zacks EA time;
586
587     proc sql; create table anntimes as select distinct a.*, b.EPS as EPS_WSH,
588 b.EA_DateTime as WSH_EA_DateTime
589     from anntimes as a left join wsh as b
590     on a.ticker=b.ticker and intck('week',a.datadate,b.QuarterEnd)=0; quit;
591
592     proc sort data=anntimes nodup;
593     by gvkey fyearqtr descending EPS_WSH;
594     run;
595
596     proc sort data=anntimes nodupkey;
597     by gvkey fyearqtr;
598     run;
599
600
601     data anntimes (drop=WSH_EA_DateTime);
602     set anntimes (keep = gvkey fyearqtr rdq ANNDATS ANNTIMS WSH_EA_DateTime);
603     WSH_ANNDATS = datepart(WSH_EA_DateTime);
604     WSH_ANNTIMS = timepart(WSH_EA_DateTime);
605     format WSH_ANNDATS date9.;
606     format WSH_ANNTIMS time.;
607     format rdq date9.;
608     run;
609
610     * get trading date of rdq and trading_date_plus1;

```

```

611
612 proc sql; create table anntimes as select distinct a.*, b.trading_date as
613 rdq_trading_date, b.trading_date_plus1 as rdq_trading_date_plus1
614 from anntimes as a left join sas.trading_dates_final as b
615 on a.rdq=b.orig_date; quit;
616
617 * next_day is trading_date, unless rdq=trading_date (in that case, next_day is
618 trading_date);
619
620 data anntimes;
621 set anntimes;
622 rdq_nextday=rdq_trading_date;
623 if rdq=rdq_trading_date then rdq_nextday=rdq_trading_date_plus1;
624 drop rdq_trading;
625 run;
626
627
628 data anntimes;
629 set anntimes;
630 format rdq_nextday date9.;
631 if rdq=WSH_ANNDATS then final_anntims=WSH_ANNTIMS;
632 if rdq=WSH_ANNDATS then final_anndats=WSH_ANNDATS;
633 if rdq=ANNDATS and rdq^=WSH_ANNDATS then final_anntims=ANNTIMS;
634 if rdq=ANNDATS and rdq^=WSH_ANNDATS then final_anndats=ANNDATS;
635 format final_anndats date9.;
636 use_nextday=0;
637 if hour(final_anntims)>15 then use_nextday=1;
638 format final_anntims time.;
639 run;
640
641
642 proc export data=anntimes
643     outfile='YOUR_DIR/anntimes.csv'
644     dbms=csv
645     replace;
646 run;
647
648 %mend anntimes;
649
650 *****;
651 ***** Sample selection (before requiring UE/depth/spread) *****;
652 *****;
653
654 %macro morevars;
655
656 * Pull Compustat variables: earnings surprise, ROA, sales, busy EA day, advertising
657 expense;
658
659 proc sort data=zacks_rdq (keep=gvkey permno cusip ticker fyearqtr datadate
660 datadate_trading comnam shrcd rdq epsfxq pre post crspshradj_qtr_end
661 crspprcadj_qtr_end crsp_price shrout zid actual_eps consensus_eps value
662 number_of_est report_date ibes_ue numest AI_cov AP_cov ret_0_2 trading_date
663 trading_date_plus1 trading_date_plus2) nodup out=determinants;
664 by gvkey fyearqtr; run;
665
666         proc sql;
667             create          table compq as
668             select          a.*, (a.fyearq * 10) + a.fqtr as fyearqtr
669             from wrds.compfundq a
670             where ((indfmt = 'INDL') and (consol = 'C') and (popsrc = 'D'))
671 and (datafmt = 'STD'))

```

```

672             order by gvkey, fyearqtr, datafqtr, atq descending, epsfxq
673 descending, ceqq descending, ltq descending;
674 quit;
675
676     * eliminate duplicate quarters - very small number so just ignore them;
677     proc sort data= compq    nodupkey;
678         by gvkey fyearqtr;
679     run;
680
681 proc sql; create table determinants as select
682 a.*, b.datafqtr, b.fyearq, b.fqtr, b.saleq, b.ibq, b.ceqq, b.atq-b.ltq as book,
683 b.atq, b.prccq, b.cshoq
684 from determinants as a left join compq (where=(indfmt='INDL' and datafmt='STD' and
685 popsrc='D' and consol='C')) as b
686 on a.gvkey=b.gvkey and a.fyearqtr=(b.fyearq*10)+b.fqtr; quit;
687
688 proc sql; create table determinants as select
689 a.*, b.epsfxq as py_eps, b.datadate as datadate_py
690 from determinants as a left join compq (where=(indfmt='INDL' and datafmt='STD' and
691 popsrc='D' and consol='C')) as b
692 on a.gvkey=b.gvkey and a.fqtr=b.fqtr and a.fyearq=b.fyearq+1; quit;
693
694 proc sql; create table determinants
695 as select a.*, b.trading_date as datadate_py_trading
696 from determinants as a left join sas.trading_dates_final as b
697 on a.datadate_py=b.orig_date; quit;
698
699 proc sql;
700     create table determinants
701     as select a.*, b.cfacsadj as crspshradj_qtr_end_py, b.cfacspr as
702 crspprcadj_qtr_end_py, abs(b.prc) as crsp_price_py
703     from determinants as a left join wrds.crspdsf as b
704     on a.permno=b.permno and a.datadate_py_trading=b.date;quit;
705
706 proc sql;
707     create table determinants
708     as select a.*, abs(b.prc)*b.shrout/1000 as crsp_mktcap
709     from determinants as a left join wrds.crspdsf as b
710     on a.permno=b.permno and a.datadate_trading=b.date;quit;
711
712 data determinants;
713 set determinants;
714 if crspshradj_qtr_end_py='' then crspshradj_qtr_end_py=1;
715 if crspprcadj_qtr_end_py='' then crspprcadj_qtr_end_py=1;
716 mkt_cap=prccq*cshoq;
717 if mkt_cap=. then mkt_cap=crsp_mktcap;
718 py_eps_adj=(py_eps/crspshradj_qtr_end_py)*crspshradj_qtr_end;
719 py_price_adj=(crsp_price_py/crspprcadj_qtr_end_py)*crspprcadj_qtr_end;
720 srw_ue=(epsfxq-py_eps_adj)/py_price_adj;
721 btm=ceqq/mkt_cap;
722 if btm=. then btm=book/mkt_cap;
723 if crsp_price='' then crsp_price=prccq;
724 loss='';
725 if actual_eps>=0 then loss=0;
726 if actual_eps<0 then loss=1;
727 if (loss='' and value>=0) then loss=0;
728 if (loss='' and value<0) then loss=1;
729 if (loss='' and epsfxq>=0) then loss=0;
730 if (loss='' and epsfxq<0) then loss=1; *indicator variable for loss quarter is
731 three tiered: first Zacks, then IBES, then compustat;
732 run;

```

```

733
734 *Adding sales over the prior year and roa;
735
736 data compdata;
737 set wrds.compfundq (keep= gvkey datafqtr datadate fyearq rdq saleq ibq atq indfmt
738 datafmt popsrc consol where=(fyearq>2007 and datafqtr ne '' and indfmt='INDL' and
739 datafmt='STD' and popsrc='D' and consol='C'));
740 run;
741
742 proc sort data=compdata; by gvkey datafqtr; run;
743
744 **Estimating busy day information and adding to dataset;
745
746 proc sql;
747     create table tempdays
748     as select distinct rdq, count(distinct gvkey) as num_eas
749     from compdata
750     where rdq ne .
751     group by rdq;quit;
752
753 proc sql;
754     create table determinants
755     as select distinct a.*, b.num_eas
756     from determinants as a left join tempdays as b
757     on a.rdq=b.rdq;quit;
758
759 *Industry;
760
761 proc sql; create table determinants as select a.*, b.sic
762 from determinants as a left join wrds.comp_company as b
763 on a.gvkey=b.gvkey; quit;
764
765 * Zacks earnings surprise;
766 * Zacks data is already adjusted for splits;
767
768 data determinants_all;
769 set determinants;
770 zacks_ue=(actual_eps-consensus_eps)/(crsp_price/crsp_prcadj_qtr_end);
771 run;
772
773 * Cleanup;
774 * cleanup and calculating variables;
775 proc sort data=determinants_all nodupkey;
776 by gvkey fyearqtr; run;
777
778 data determinants_final (keep=gvkey datadate datadate_trading permno cusip ticker
779 rdq comnam fyearqtr fyearq fqtr AI_cov AP_cov pre post loss btm mkt_cap lnmv
780 ln_analysts number_of_est report_date ibes_ue zacks_ue zid ue srw_ue num_eas sic
781 crsp_price);
782 set determinants_all;
783 lnmv=log(1+mkt_cap);
784 if number_of_est='' then number_of_est=0;
785 ln_zacks_analysts=log(1+number_of_est);
786 if numest='' then numest=0;
787 ln_ibes_analysts=log(1+numest);
788 ln_analysts=max(ln_zacks_analysts,ln_ibes_analysts);
789 ue=mean(zacks_ue,ibes_ue);
790 run;
791
792 data determinants_final (drop=sic calqtr);
793 set determinants_final;

```

```

794 calqtr=1;
795 if (month(rdq)>3 and month(rdq)<=6) then calqtr=2;
796 if (month(rdq)>6 and month(rdq)<=9) then calqtr=3;
797 if (month(rdq)>9 and month(rdq)<=12) then calqtr=4;
798 cyearqtr=cats(year(rdq),calqtr);
799 yearmonth=cats(year(rdq),month(rdq));
800 year=year(rdq);
801 run;
802
803 data all_data;
804 set zacks_rdq;
805 calqtr=1;
806 if (month(rdq)>3 and month(rdq)<=6) then calqtr=2;
807 if (month(rdq)>6 and month(rdq)<=9) then calqtr=3;
808 if (month(rdq)>9 and month(rdq)<=12) then calqtr=4;
809 cyearqtr=cats(year(rdq),calqtr);
810 run;
811
812 proc sort data=all_data (keep=gvkey fyearqtr cyearqtr AI_cov) nodup
813 out=conditionalAI;
814 by gvkey cyearqtr; run;
815
816 data conditionalAI;
817 set conditionalAI;
818 by gvkey;
819 after_AI_init=1;
820 if first.gvkey then sum_AI=0;
821     sum_AI+AI_cov;
822 if sum_AI>0 then output;
823 run;
824
825 * Merge gvkey fyearqtr in conditionalAI into determinants_final to get the post-AI
826 initiation quarters only;
827 * indicator variable for post-AI;
828
829 proc sql; create table determinants_final as select
830 a.*, b.after_AI_init
831 from determinants_final as a left join conditionalAI as b
832 on a.gvkey=b.gvkey and a.fyearqtr=b.fyearqtr; quit;
833
834 data determinants_final;
835 set determinants_final;
836 if after_AI_init=' ' then after_AI_init=0;
837 run;
838
839 %mend morevars;
840
841 *****;
842 ***** Spreads and Depths *****;
843 *****;
844
845 %macro spreadsdepths;
846
847 * 1. get dates for pre-event window;
848 *the pre-event window is (-41,-11) trading days relative to the EA;
849
850 proc sql; create table spread as select
851 a.*, b.rdq as prev_rdq
852 from determinants_final as a left join wrds.compfundq (where=(datafqtr^=' ')) as b
853 on (a.gvkey=b.gvkey and ((a.fyearq=b.fyearq and a.fqtr>1 and a.fqtr=b.fqtr+1) or
854     (a.fyearq=b.fyearq+1 and a.fqtr=1 and b.fqtr=4)

```

```

855         )); quit;
856
857 proc sort data=spread;
858 by gvkey fyearqtr descending prev_rdq; run;
859
860 proc sort data=spread nodupkey;
861 by gvkey fyearqtr; run;
862
863 data spread;
864 set spread;
865 betw_qtrs=intck('weekday',prev_rdq,rdq);
866 run;
867
868 proc sql; create table spread as select
869 a.*, b.trading_date_plus3 as prev_rdq_plus3
870 from spread as a left join sas.trading_dates_final as b
871 on a.prev_rdq=b.orig_date; quit;
872
873 proc sql; create table spread as select
874 a.*, b.trading_date_min2 as rdq_min2
875 from spread as a left join sas.trading_dates_final as b
876 on a.rdq=b.orig_date; quit;
877
878 proc sql; create table spread as select
879 a.*, b.trading_date_min60 as prev_rdq_plus3_alt
880 from spread as a left join sas.trading_dates_final as b
881 on a.rdq_min2=b.orig_date; quit;
882
883 data spread (drop=prev_rdq prev_rdq_plus3_alt betw_qtrs);
884 set spread;
885 if (prev_rdq_plus3='' or betw_qtrs<10) then prev_rdq_plus3=prev_rdq_plus3_alt;
886 run;
887
888 ** merge in correct announcement date;
889
890 proc sql; create table spread as select distinct a.*, b.rdq_nextday, b.use_nextday
891 from spread as a left join anntimes as b on a.gvkey=b.gvkey and
892 a.fyearqtr=b.fyearqtr; quit;
893
894 data spread;
895 set spread;
896 rdq_to_use=rdq;
897 if use_nextday=1 then rdq_to_use=rdq_nextday;
898 drop rdq_nextday use_nextday;
899 run;
900
901 proc sql; create table spread1 as select a.*, b.trading_date, b.trading_date_plus1,
902 b.trading_date_plus2,b.trading_date_plus3,b.trading_date_plus4,b.trading_date_plus5
903 , b.trading_date_min11 as rdq_min11, b.trading_date_min41 as rdq_min41
904 from spread as a left join sas.trading_dates_final as b
905 on a.rdq_to_use=b.orig_date; quit;
906
907 * indicator variable for last observation before post period, so we can do
908 descriptives of firm characteristics;
909
910 proc sort data=spread1 out=last_pre;
911 by gvkey descending fyearqtr; run;
912
913 data last_pre;
914 set last_pre;
915 if rdq<'13oct2014'd;

```

```

916 run;
917
918 data last_pre;
919 set last_pre;
920 by gvkey;
921 if first.gvkey then output;
922 run;
923
924 proc sql; create table spread1
925 as select a.*, b.gvkey as match_lastpre
926 from spread1 as a left join last_pre as b
927 on a.gvkey=b.gvkey and a.fyearqtr=b.fyearqtr; quit;
928
929 data spread1 (drop=match_lastpre);
930 set spread1;
931 last_pre=0;
932 if match_lastpre^='' then last_pre=1;
933 run;
934
935 *****;
936 ** CONTROLS: abn_turn;
937 *****;
938
939 proc sort data=spread1 nodupkey;
940 by gvkey fyearqtr; run;
941
942 * get abn_turn;
943
944 proc sql; create table spread1
945 as select distinct a.*, mean(b.vol/(1000*b.shrout)) as turn_event_firm
946 from spread1 as a left join wrds.crspdsf (where=(vol>=0)) as b
947 on a.permno=b.permno and intck('day',a.trading_date,b.date)>=0 and
948 intck('day',a.trading_date_plus2,b.date)<=0
949 group by a.gvkey, a.fyearqtr;
950 quit;
951
952 proc sql; create table spread1
953 as select distinct a.*, mean(b.vol/(1000*b.shrout)) as turn_event_firm_0_1
954 from spread1 as a left join wrds.crspdsf (where=(vol>=0)) as b
955 on a.permno=b.permno and intck('day',a.trading_date,b.date)>=0 and
956 intck('day',a.trading_date_plus1,b.date)<=0
957 group by a.gvkey, a.fyearqtr;
958 quit;
959
960 proc sql; create table spread1
961 as select distinct a.*, mean(b.vol/(1000*b.shrout)) as turn_event_firm_0_3
962 from spread1 as a left join wrds.crspdsf (where=(vol>=0)) as b
963 on a.permno=b.permno and intck('day',a.trading_date,b.date)>=0 and
964 intck('day',a.trading_date_plus3,b.date)<=0
965 group by a.gvkey, a.fyearqtr;
966 quit;
967
968 proc sql; create table spread1
969 as select distinct a.*, mean(b.vol/(1000*b.shrout)) as turn_event_firm_0_4
970 from spread1 as a left join wrds.crspdsf (where=(vol>=0)) as b
971 on a.permno=b.permno and intck('day',a.trading_date,b.date)>=0 and
972 intck('day',a.trading_date_plus4,b.date)<=0
973 group by a.gvkey, a.fyearqtr;
974 quit;
975
976 proc sql; create table spread1

```



```

977 as select distinct a.*, mean(b.vol/(1000*b.shrout)) as turn_event_firm_0_5
978 from spread1 as a left join wrds.crspdsf (where=(vol>=0)) as b
979 on a.permno=b.permno and intck('day',a.trading_date,b.date)>=0 and
980 intck('day',a.trading_date_plus5,b.date)<=0
981 group by a.gvkey, a.fyearqtr;
982 quit;
983
984 proc sql; create table spread1
985 as select distinct a.*, mean(b.vol/(1000*b.shrout)) as turn_tavg_firm
986 from spread1 as a left join wrds.crspdsf (where=(vol>=0)) as b
987 on a.permno=b.permno and intck('day',a.rdq_min41,b.date)>=0 and
988 intck('day',a.rdq_min11,b.date)<=0
989 group by a.gvkey, a.fyearqtr;
990 quit;
991
992 * volatility;
993
994 proc sql; create table spread1
995 as select distinct a.*, std(1+b.ret)*sqrt(252) as volat_pre
996 from spread1 as a left join wrds.crspdsf as b
997 on a.permno=b.permno and intck('day',a.prev_rdq_plus3,b.date)>=0 and
998 intck('day',a.rdq_min2,b.date)<=0
999 group by a.gvkey, a.fyearqtr;
1000 quit;
1001
1002 * mkt turnover;
1003
1004 proc sql; create table mkt_turn as select distinct
1005 date, sum(vol) as dailytotvol, sum(shrout) as dailytotshrout
1006 from wrds.crspdsf (where=(vol>=0)) group by date;
1007 quit;
1008
1009 proc sql; create table spread1
1010 as select distinct a.*, mean(b.dailytotvol/(1000*b.dailytotshrout)) as
1011 turn_event_mkt
1012 from spread1 as a left join mkt_turn as b
1013 on intck('day',a.trading_date,b.date)>=0 and
1014 intck('day',a.trading_date_plus2,b.date)<=0
1015 group by a.gvkey, a.fyearqtr;
1016 quit;
1017
1018 proc sql; create table spread1
1019 as select distinct a.*, mean(b.dailytotvol/(1000*b.dailytotshrout)) as
1020 turn_event_mkt_0_1
1021 from spread1 as a left join mkt_turn as b
1022 on intck('day',a.trading_date,b.date)>=0 and
1023 intck('day',a.trading_date_plus1,b.date)<=0
1024 group by a.gvkey, a.fyearqtr;
1025 quit;
1026
1027 proc sql; create table spread1
1028 as select distinct a.*, mean(b.dailytotvol/(1000*b.dailytotshrout)) as
1029 turn_event_mkt_0_3
1030 from spread1 as a left join mkt_turn as b
1031 on intck('day',a.trading_date,b.date)>=0 and
1032 intck('day',a.trading_date_plus3,b.date)<=0
1033 group by a.gvkey, a.fyearqtr;
1034 quit;
1035
1036 proc sql; create table spread1

```

```

1037 as select distinct a.*, mean(b.dailytotvol/(1000*b.dailytotshrout)) as
1038 turn_event_mkt_0_4
1039 from spread1 as a left join mkt_turn as b
1040 on intck('day',a.trading_date,b.date)>=0 and
1041 intck('day',a.trading_date_plus4,b.date)<=0
1042 group by a.gvkey, a.fyearqtr;
1043 quit;
1044
1045 proc sql; create table spread1
1046 as select distinct a.*, mean(b.dailytotvol/(1000*b.dailytotshrout)) as
1047 turn_event_mkt_0_5
1048 from spread1 as a left join mkt_turn as b
1049 on intck('day',a.trading_date,b.date)>=0 and
1050 intck('day',a.trading_date_plus5,b.date)<=0
1051 group by a.gvkey, a.fyearqtr;
1052 quit;
1053 proc sql; create table spread1
1054 as select distinct a.*, mean(b.dailytotvol/(1000*b.dailytotshrout)) as
1055 turn_tavg_mkt
1056 from spread1 as a left join mkt_turn as b
1057 on intck('day',a.rdq_min41,b.date)>=0 and intck('day',a.rdq_min11,b.date)<=0
1058 group by a.gvkey, a.fyearqtr;
1059 quit;
1060
1061 data spread1 (keep=gvkey permno cusip ticker datadate datadate_trading rdq
1062 rdq_to_use fyearqtr cyearqtr reporting_lag AI_cov AP_cov mkt_cap post pre yearmonth
1063 year comnam num_eas ibes_ue zacks_ue zid ue srw_ue loss lnmv btm ln_analysts
1064 number_of_est report_date abn_turn_event abn_turn_event_0_1 abn_turn_event_0_3
1065 abn_turn_event_0_4 abn_turn_event_0_5 turn_event_firm volat_pre crsp_price);
1066 set spread1;
1067 abn_turn_pre=turn_tavg_firm-turn_tavg_mkt;
1068 abn_turn_event=100*((turn_event_firm-turn_event_mkt)-abn_turn_pre);
1069 abn_turn_event_0_1=100*((turn_event_firm_0_1-turn_event_mkt_0_1)-abn_turn_pre);
1070 abn_turn_event_0_3=100*((turn_event_firm_0_3-turn_event_mkt_0_3)-abn_turn_pre);
1071 abn_turn_event_0_4=100*((turn_event_firm_0_4-turn_event_mkt_0_4)-abn_turn_pre);
1072 abn_turn_event_0_5=100*((turn_event_firm_0_5-turn_event_mkt_0_5)-abn_turn_pre);
1073 turnover=100*(turn_event_firm-turn_event_mkt);
1074 reporting_lag=log(datdif(datadate, rdq, 'ACT/ACT'));
1075 run;
1076
1077 * add 4th fiscal quarter indicator;
1078
1079 data spread1;
1080 set spread1;
1081 fqtr_4=0;
1082 if substrn(fyearqtr,5,1)=4 then fqtr_4=1;
1083 run;
1084
1085 *****;
1086 * restrict to observations (firm-quarters) without missing data;
1087 data spread2;
1088 set spread1 (keep=gvkey datadate datadate_trading rdq rdq_to_use fyearqtr permno
1089 cusip ticker AI_cov AP_cov post pre mkt_cap btm loss lnmv ln_analysts number_of_est
1090 report_date num_eas ibes_ue zacks_ue zid ue srw_ue cyearqtr year yearmonth
1091 volat_pre abn_turn_event abn_turn_event_0_1 abn_turn_event_0_3 abn_turn_event_0_4
1092 abn_turn_event_0_5 turn_event_firm reporting_lag comnam fqtr_4 crsp_price);
1093 ue_either=ue;
1094 if ue_either='' then ue_either=srw_ue;
1095 if cmiss(gvkey, datadate, rdq, fyearqtr, permno, AI_cov, AP_cov, post, pre,
1096 mkt_cap, btm, loss, lnmv, ln_analysts, num_eas, ue_either, cyearqtr, year,
1097 yearmonth, volat_pre, abn_turn_event, turn_event_firm, reporting_lag, comnam,

```

```

1098 fqtr_4, crsp_price)=0; * complete cases for all vars (except only complete cases
1099 of ue_either, so either analyst or srw ue is needed, not both);
1100 year=year(rdq);
1101 run;
1102
1103 proc sort data=spread2 (drop=ue_either) nodup;
1104 by gvkey fyearqtr; run;
1105
1106 * quantiles of busy_ea day, signed earnings surprise (9 is highest number of ann, 0
1107 is lowest);
1108 proc sort data=spread2;
1109 by year; run;
1110
1111 proc rank data=spread2 out=spread2 ties=low groups=10;
1112 by year;
1113 var num_eas ue;
1114 ranks busyday analyst_ue_rank;
1115 run;
1116
1117 proc sort data=spread2;
1118 by year analyst_ue_rank; run;
1119
1120 proc rank data=spread2 out=spread2 ties=low groups=10;
1121 by year analyst_ue_rank; *rank within year and only for firms without analysts;
1122 var srw_ue;
1123 ranks srw_ue_rank;
1124 run;
1125
1126 data spread2 (drop = srw_ue_rank analyst_ue_rank);
1127 set spread2;
1128 ue_rank=analyst_ue_rank;
1129 if ue_rank=' ' then ue_rank=srw_ue_rank;
1130 run;
1131
1132
1133 *****;
1134 * restrict to most recent fyearqtr for duplicate firm-rdq obs;
1135
1136 proc sort data=spread2;
1137 by gvkey rdq_to_use descending fyearqtr; run;
1138
1139 proc sort data=spread2 nodupkey;
1140 by gvkey rdq_to_use; run;
1141
1142 *****;
1143 * restrict to firms with at least one pre and post observation;
1144
1145 proc sort data=spread2 nodup;
1146 by gvkey fyearqtr; run;
1147
1148
1149 data zacks_rdq1_hand;
1150 set spread2;
1151 by gvkey;
1152 if first.gvkey then sum_pre=0;
1153 if first.gvkey then sum_post=0;
1154 sum_pre+pre;
1155 sum_post+post;
1156 if last.gvkey then output;
1157 run;
1158

```

```

1159 data zacks_prepost_hand;
1160 set zacks_rdql_hand;
1161 if (sum_pre>0 and sum_post>0);
1162 run;
1163
1164 * match pre_post indicator into firm-qtr data;
1165
1166 proc sql; create table spread2 as select distinct
1167 a.*, b.gvkey as pre_post_gvkey
1168 from spread2 as a left join zacks_prepost_hand as b
1169 on a.gvkey=b.gvkey; quit;
1170
1171 data spread2 (drop = pre_post_gvkey);
1172 set spread2;
1173 if pre_post_gvkey^='';
1174 run;
1175
1176 proc sort data=spread2 nodupkey;
1177 by gvkey fyearqtr; run;
1178
1179 %mend spreadsdepths;
1180
1181 *****;
1182 ** further sample selection through descriptives steps;
1183 *****;
1184
1185 %macro descriptives;
1186
1187 * 1. All firms - available sample 60,566;
1188
1189 proc sql; create table spread_2 as select a.*, b.after_AI_init
1190 from spread2 as a left join determinants_final as b
1191 on a.gvkey=b.gvkey and a.fyearqtr=b.fyearqtr; quit;
1192
1193 data pre;
1194 set spread_2 (where=(after_AI_init=0));
1195 by gvkey;
1196 if first.gvkey then sum_quarters=0;
1197 if first.gvkey then AP_cov_quarter=0;
1198     sum_quarters+pre;
1199     AP_cov_quarter+AP_cov;
1200 if last.gvkey then output;
1201 run;
1202
1203 data pre;
1204 set pre;
1205 AP_cov_pct=AP_cov_quarter/sum_quarters;
1206 if AP_cov_pct=0;
1207 run;
1208
1209 proc sql; create table post as select a.*, b.after_AI_init
1210 from spread2 (where=(post=1)) as a left join determinants_final as b
1211 on a.gvkey=b.gvkey and a.fyearqtr=b.fyearqtr; quit;
1212
1213 data post;
1214 set post (where=(after_AI_init=1));
1215 by gvkey;
1216 if first.gvkey then sum_quarters=0;
1217 if first.gvkey then AI_cov_quarter=0;
1218     sum_quarters+after_AI_init;
1219     AI_cov_quarter+AI_cov;

```

```

1220 if last.gvkey then output;
1221 run;
1222
1223 data post;
1224 set post;
1225 missing_AI_quarter=sum_quarters-AI_cov_quarter;
1226 if missing_AI_quarter=1;
1227 run;
1228
1229 proc sql; create table spread2 as select a.*, b.gvkey as expanded_treat_gvkey
1230 from spread2 as a left join pre as b
1231 on a.gvkey=b.gvkey; quit;
1232
1233 proc sql; create table spread2 as select a.*, b.gvkey as expanded_treat_gvkey2
1234 from spread2 as a left join post as b
1235 on a.gvkey=b.gvkey; quit;
1236
1237 proc sort data=spread2 (keep=gvkey rdq fyearqtr cyearqtr AI_cov AP_cov mkt_cap)
1238 nodup out=consistency2;
1239 by gvkey cyearqtr; run;
1240
1241 proc sql; create table consistency2
1242 as select a.*, b.after_AI_init
1243 from consistency2 as a left join determinants_final as b
1244 on a.gvkey=b.gvkey and a.fyearqtr=b.fyearqtr; quit;
1245
1246 data any_ai_init;
1247 set determinants_final;
1248 if after_AI_init=1;
1249 run;
1250
1251 proc sort data=any_ai_init nodupkey; by gvkey; run;
1252
1253 proc sql; create table consistency2
1254 as select distinct a.*, b.after_AI_init as any_ai_init
1255 from consistency2 as a left join any_ai_init as b
1256 on a.gvkey=b.gvkey; quit;
1257
1258 data consistency2;
1259 set consistency2;
1260 if ((after_AI_init=0 and any_ai_init=1) or (rdq<'13oct2014'd and any_ai_init=''));
1261 run;
1262
1263 proc sort data=consistency2; by gvkey; run;
1264
1265 data consistency2;
1266 set consistency2;
1267 by gvkey;
1268 aquarter=1;
1269 if first.gvkey then sum_AP=0;
1270 if first.gvkey then sum_quarters=0;
1271     sum_AP+AP_cov;
1272     sum_quarters+aquarter;
1273 if last.gvkey then output;
1274 run;
1275
1276 data consistency2 (keep=gvkey mkt_cap pct_covered);
1277 set consistency2;
1278 pct_covered=sum_AP/sum_quarters;
1279 run;
1280

```

```

1281 proc sql; create table spread_descriptives as select distinct a.*, b.pct_covered as
1282 pre_AP_covpct
1283 from spread2 as a left join consistency2 as b
1284 on a.gvkey=b.gvkey; quit;
1285
1286 proc sort data=spread2 (keep=gvkey rdq fyearqtr cyearqtr AI_cov AP_cov mkt_cap)
1287 nodup out=consistency2;
1288 by gvkey cyearqtr; run;
1289
1290 proc sql; create table consistency2
1291 as select a.*, b.after_AI_init
1292 from consistency2 as a left join determinants_final as b
1293 on a.gvkey=b.gvkey and a.fyearqtr=b.fyearqtr; quit;
1294
1295 proc sql; create table consistency2
1296 as select distinct a.*, b.after_AI_init as any_ai_init
1297 from consistency2 as a left join any_ai_init as b
1298 on a.gvkey=b.gvkey; quit;
1299
1300 data consistency2;
1301 set consistency2;
1302 if ((after_AI_init=1 and any_ai_init=1) or (rdq>='13oct2014'd and any_ai_init=''));
1303 run;
1304
1305 proc sort data=consistency2; by gvkey; run;
1306
1307 data consistency2;
1308 set consistency2;
1309 by gvkey;
1310 aquarter=1;
1311 if first.gvkey then sum_AP=0;
1312 if first.gvkey then sum_AI=0;
1313 if first.gvkey then sum_quarters=0;
1314     sum_AP+AP_cov;
1315     sum_AI+AI_cov;
1316     sum_quarters+aquarter;
1317 if last.gvkey then output;
1318 run;
1319
1320 data consistency2 (keep=gvkey mkt_cap pct_covered pct_covered_AI);
1321 set consistency2;
1322 pct_covered=sum_AP/sum_quarters;
1323 pct_covered_AI=sum_AI/sum_quarters;
1324 run;
1325
1326 proc sql; create table spread_descriptives as select distinct a.*, b.pct_covered as
1327 post_AP_covpct, b.pct_covered_AI as post_AI_covpct
1328 from spread_descriptives as a left join consistency2 as b
1329 on a.gvkey=b.gvkey; quit;
1330
1331 proc sort data=spread_descriptives nodupkey;
1332 by gvkey; run;
1333
1334 proc sql; create table spread_descriptives
1335 as select distinct a.*, b.after_AI_init as any_ai_init
1336 from spread_descriptives as a left join any_ai_init as b
1337 on a.gvkey=b.gvkey; quit;
1338
1339 data spread_descriptives_test;
1340 set spread_descriptives;
1341 if any_ai_init='';

```

```

1342 run;
1343
1344 * 2. Sample of Non-Adopters and Adopters with Full AI Coverage in Post Period;
1345
1346 proc sort data=spread2 (keep=gvkey cyearqtr AI_cov) nodup out=consistency;
1347 by gvkey cyearqtr; run;
1348
1349 data consistency;
1350 set consistency;
1351 by gvkey;
1352 after_AI_init=1;
1353 if first.gvkey then sum_AI=0;
1354     sum_AI+AI_cov;
1355 if sum_AI>0 then output;
1356 run;
1357
1358 data consistency;
1359 set consistency;
1360 by gvkey;
1361 if first.gvkey then sum_quarters=0;
1362     sum_quarters+after_AI_init;
1363 if last.gvkey then output;
1364 run;
1365
1366 data consistency (keep=gvkey pct_covered);
1367 set consistency;
1368 pct_covered=sum_AI/sum_quarters;
1369 run;
1370
1371 data consistency_100;
1372 set consistency;
1373 if pct_covered=1;
1374 run;
1375
1376 * consistency_100 has firms that are 100% covered by AI in the post period;
1377 * expanded_treat_gvkey2 has the 1 skipped quarter AI in the post period firms;
1378
1379 proc sql; create table spread_descriptives as select distinct a.*, b.gvkey as
1380 expanded_treat_100
1381 from spread_descriptives as a left join consistency_100 as b
1382 on a.gvkey=b.gvkey; quit;
1383
1384 data spread_descriptives2;
1385 set spread_descriptives;
1386 if (expanded_treat_100^='' or expanded_treat_gvkey2^='' or post_AI_covpct=0);
1387 run;
1388
1389 proc contents data=spread_descriptives2; run;
1390
1391 data spread_descriptives_test;
1392 set spread_descriptives2;
1393 if any_ai_init='';
1394 run;
1395
1396 proc sql; create table spread_descriptives_test as select distinct a.*,
1397 b.post_AI_covpct
1398 from spread2 as a left join spread_descriptives as b
1399 on a.gvkey=b.gvkey; run;
1400
1401 proc sql; create table spread_descriptives_test as select distinct a.*, b.gvkey as
1402 expanded_treat_100

```

```

1403 from spread_descriptives_test as a left join consistency_100 as b
1404 on a.gvkey=b.gvkey; quit;
1405
1406 proc sql; create table spread_descriptives_test as select a.*, b.after_AI_init
1407 from spread_descriptives_test as a left join determinants_final as b
1408 on a.gvkey=b.gvkey and a.fyearqtr=b.fyearqtr; quit;
1409
1410 data spread_descriptives_test;
1411 set spread_descriptives_test;
1412 if (expanded_treat_100^='' or expanded_treat_gvkey2^='' or post_AI_covpct=0);
1413 run;
1414
1415 data spread_descriptives_test;
1416 set spread_descriptives_test;
1417 if (expanded_treat_gvkey2^='' and after_AI_init=1 and AI_cov=0) then delete;
1418 run;
1419
1420 proc sql; create table spread2 as select distinct a.*, b.pre_AP_covpct,
1421 b.post_AP_covpct, b.post_AI_covpct
1422 from spread2 as a left join spread_descriptives as b
1423 on a.gvkey=b.gvkey; quit;
1424
1425 proc sql; create table spread2 as select a.*, b.after_AI_init
1426 from spread2 as a left join determinants_final as b
1427 on a.gvkey=b.gvkey and a.fyearqtr=b.fyearqtr; quit;
1428
1429 * get c2 (control 2 - firms getting 0 AP cov in pre period and 0% post period AI or
1430 AP cov);
1431
1432 data spread2 (drop = control2 expanded_treat_gvkey expanded_treat_gvkey2
1433 post_AP_covpct);
1434 set spread2;
1435 treat=0;
1436 if (pre_AP_covpct=0 and (post_AI_covpct=1 or expanded_treat_gvkey2^='')) then
1437 treat=1;
1438 if (expanded_treat_gvkey2^='' and after_AI_init=1 and AI_cov=0) then treat=0;
1439 control2=0;
1440 if (pre_AP_covpct=0 and post_AI_covpct=0 and post_AP_covpct=0) then control2=1;
1441 post_AIinit=0;
1442 if (treat=1 and AI_cov=1) then post_AIinit=1;
1443 if control2=1 then post_AIinit=post;
1444 run;
1445
1446 *count number of AI quarters;
1447
1448 data count_AI_qtrs;
1449 set spread2 (keep=gvkey fyearqtr rdq AI_cov post_AIinit);
1450 run;
1451
1452 proc sort data=count_AI_qtrs (where=(post_AIinit=1));
1453 by gvkey fyearqtr; run;
1454
1455 data count_AI_qtrs;
1456 set count_AI_qtrs;
1457 by gvkey;
1458 if first.gvkey then num_AI_qtrs=0; *make sure this sets the first AI quarter to 1;
1459 num_AI_qtrs+AI_cov;
1460 run;
1461
1462 data count_AI_qtrs;
1463 set count_AI_qtrs;

```



```

1464 AI_Q1=0;
1465 if num_AI_qtrs=1 then AI_Q1=1;
1466 AI_Q2=0;
1467 if num_AI_qtrs=2 then AI_Q2=1;
1468 AI_Q3=0;
1469 if num_AI_qtrs=3 then AI_Q3=1;
1470 AI_Q4=0;
1471 if num_AI_qtrs=4 then AI_Q4=1;
1472 AI_Q5=0;
1473 if num_AI_qtrs=5 then AI_Q5=1;
1474 run;
1475
1476 proc sql; create table spread2 as
1477 select a.*, b.AI_Q1, b.AI_Q2, b.AI_Q3, b.AI_Q4, b.AI_Q5
1478 from spread2 as a left join count_AI_qtrs as b
1479 on a.gvkey=b.gvkey and a.fyearqtr=b.fyearqtr; quit;
1480
1481 data spread2 (drop = post_AIinit);
1482 set spread2;
1483 if AI_Q1='' then AI_Q1=0;
1484 if AI_Q2='' then AI_Q2=0;
1485 if AI_Q3='' then AI_Q3=0;
1486 if AI_Q4='' then AI_Q4=0;
1487 if AI_Q5='' then AI_Q5=0;
1488 run;
1489
1490 ** spread3 has the ranked variables based on the Sample of Non-Adopters and
1491 Adopters with Full AI Coverage in Post Period;
1492
1493 proc sql; create table spread3 as select a.*, b.gvkey as ai_nonadopt_gvkey
1494 from spread2 as a left join spread_descriptives_test as b
1495 on a.gvkey=b.gvkey and a.fyearqtr=b.fyearqtr; quit;
1496
1497 data spread3 (drop= ai_nonadopt_gvkey busyday ue_rank);
1498 set spread3;
1499 if ai_nonadopt_gvkey^='';
1500 run;
1501
1502 * re-rank;
1503
1504 proc sort data=spread3;
1505 by year; run;
1506
1507 proc rank data=spread3 out=spread3 ties=low groups=10;
1508 by year;
1509     var num_eas ue;
1510     ranks busyday analyst_ue_rank;
1511 run;
1512
1513 proc sort data=spread3;
1514 by year analyst_ue_rank; run;
1515
1516 proc rank data=spread3 out=spread3 ties=low groups=10;
1517 by year analyst_ue_rank; *rank within year and only for firms without analysts;
1518     var srw_ue;
1519     ranks srw_ue_rank;
1520 run;
1521
1522 data spread3 (drop = srw_ue_rank analyst_ue_rank);
1523 set spread3;
1524 ue_rank=analyst_ue_rank;

```

```

1525 if ue_rank='' then ue_rank=srw_ue_rank;
1526 run;
1527
1528 ** spread4 has the final sample of low coverage firms;
1529
1530 data spread4 (drop = pre_AP_covpct);
1531 set spread3;
1532 if pre_AP_covpct=0;
1533 run;
1534
1535 * 2. non-adopters with AP but no AI coverage in the post period;
1536
1537 proc sql; create table spread4 as select distinct a.*, b.after_ai_init as
1538 any_ai_init
1539 from spread4 as a left join any_ai_init as b
1540 on a.gvkey=b.gvkey; quit;
1541
1542 data test;
1543 set spread4;
1544 if (any_ai_init='' and post=1 and AP_cov=1);
1545 run;
1546
1547 proc sql; create table spread4 as select distinct a.*, b.gvkey as
1548 nonadopt_APcov_gvkey
1549 from spread4 as a left join test as b
1550 on a.gvkey=b.gvkey; quit;
1551
1552 data spread4 (drop = nonadopt_APcov_gvkey);
1553 set spread4;
1554 if nonadopt_APcov_gvkey='';
1555 run;
1556
1557 ** spread4 has the final sample of low coverage firms;
1558
1559 data spread4 (drop= busyday ue_rank);
1560 set spread4;
1561 run;
1562
1563 * re-rank;
1564
1565 proc sort data=spread4;
1566 by year; run;
1567
1568 proc rank data=spread4 out=spread4 ties=low groups=10;
1569 by year;
1570     var num_eas ue;
1571     ranks busyday analyst_ue_rank;
1572 run;
1573
1574 proc sort data=spread4;
1575 by year analyst_ue_rank; run;
1576
1577 proc rank data=spread4 out=spread4 ties=low groups=10;
1578 by year analyst_ue_rank; *rank within year and only for firms without analysts;
1579     var srw_ue;
1580     ranks srw_ue_rank;
1581 run;
1582
1583 data spread4 (drop = srw_ue_rank after_AI_init any_ai_init);
1584 set spread4;
1585 ue_rank=analyst_ue_rank;

```

```

1586 if ue_rank='' then ue_rank=srw_ue_rank;
1587 run;
1588
1589 %mend descriptives;
1590
1591 *****;
1592 ** Ravenpack variables;
1593 *****;
1594
1595 %macro raven;
1596
1597 proc sql; create table raven
1598 as select distinct a.*, count(unique b.rp_story_id) as dow_full_count
1599 from spread4 as a left join raven.matched_earnings (where=(edition='Dow' and
1600 news_type='FULL-ARTICLE')) as b
1601 on a.gvkey=b.gvkey and a.fyearqtr=b.fyearqtr
1602 group by a.gvkey, b.fyearqtr; quit;
1603
1604 proc sql; create table raven
1605 as select distinct a.*, count(unique b.rp_story_id) as dow_flash_count
1606 from raven as a left join raven.matched_earnings (where=(edition='Dow' and
1607 (news_type='HOT-NEWS-FLASH' or news_type='NEWS-FLASH')) as b
1608 on a.gvkey=b.gvkey and a.fyearqtr=b.fyearqtr
1609 group by a.gvkey, b.fyearqtr; quit;
1610
1611 data raven (drop = dow_full_count);
1612 set raven;
1613 any_dow_full=0;
1614 if dow_full_count>0 then any_dow_full=1;
1615 run;
1616
1617 * merge in quarter of AI adoption (99999 if non-adopter);
1618
1619 proc sort data=spread2 (keep=gvkey permno rdq cyearqtr AI_cov where=(AI_cov=1))
1620 nodup out=rollout;
1621 by gvkey cyearqtr; run;
1622
1623 data rollout;
1624 set rollout;
1625 by gvkey;
1626 if (cyearqtr='20151' and rdq>='14jan2015'd) then reason_later_init=1; *26 of
1627 initiations in 2015 have rdq in first two wks of jan;
1628 if first.gvkey then output;
1629 run;
1630
1631 proc sort data=spread2 (keep=gvkey permno rdq cyearqtr AI_cov post where=(post=1))
1632 nodup out=first_post_rdq;
1633 by gvkey rdq; run;
1634
1635 data first_post_rdq;
1636 set first_post_rdq;
1637 by gvkey;
1638 if first.gvkey then output;
1639 run;
1640
1641 proc sql; create table rollout1 as select
1642 a.*, b.rdq as first_post_rdq
1643 from rollout as a left join first_post_rdq as b
1644 on a.gvkey=b.gvkey; quit;
1645
1646 data rollout_truelate;

```

```

1647 set rollout1;
1648 if rdq>first_post_rdq;
1649 run;
1650
1651 data rollout_notlate;
1652 set rollout1;
1653 if rdq=first_post_rdq;
1654 run;
1655
1656 proc sql; create table raven as select a.*, b.cyearqtr as cyearqtr_early
1657 from raven as a left join rollout_notlate as b
1658 on a.gvkey=b.gvkey; quit;
1659
1660 proc sql; create table raven as select a.*, b.cyearqtr as cyearqtr_late
1661 from raven as a left join rollout_truelate as b
1662 on a.gvkey=b.gvkey; quit;
1663
1664 proc sort data=raven nodupkey;
1665 by gvkey fyearqtr; run;
1666
1667 data raven (drop=cyearqtr_early cyearqtr_late);
1668 set raven;
1669 cyearqtr_adopt=cyearqtr_early;
1670 if cyearqtr_adopt='' then cyearqtr_adopt=cyearqtr_late;
1671 *non-adopters have an empty cyearqtr_adopt field;
1672 if cyearqtr_adopt='' then cyearqtr_adopt='99999';
1673 run;
1674
1675
1676 * add friday indicator;
1677
1678 data raven1;
1679 set raven;
1680 friday=0;
1681 if weekday(rdq)=6 then friday=1;
1682 log_dowflash_count=log(1+dow_flash_count);
1683 loss_new = input(loss, 8.);
1684 drop loss;
1685 rename loss_new=loss;
1686 run;
1687
1688 %mend raven;
1689
1690 *****;
1691 ** retail and MF volume;
1692
1693 %macro volume;
1694
1695 ** add retail volume;
1696
1697 data trades_2003to2017;
1698 set taq.trades_2003to2017;
1699 keep permno date retail_volume buy_volume sell_volume retail_buy_volume
1700 retail_sell_volume;
1701 run;
1702
1703 proc sql; create table trades_2003to2017 as select distinct a.*, b.shrout
1704 from trades_2003to2017 as a left join wrds.crspdsf (where=(vol>=0)) as b
1705 on a.permno=b.permno and intck('day',a.date,b.date)=0; quit;
1706
1707

```

```

1708 proc sql; create table out_raven1 as select distinct a.*, b.trading_date,
1709 b.trading_date_plus1, b.trading_date_plus2, b.trading_date_plus3,
1710 b.trading_date_plus4, b.trading_date_plus5, b.trading_date_min41,
1711 b.trading_date_min11
1712 from raven1 as a left join sas.trading_dates_final as b
1713 on a.rdq_to_use=b.orig_date; quit;
1714
1715 proc sql; create table out_raven1
1716 as select distinct a.*, mean(b.retail_volume/(1000*b.shrout)) as turn_retail_firm
1717 from out_raven1 as a left join trades_2003to2017 (where=(shrout>0)) as b
1718 on a.permno=b.permno and intck('day',a.trading_date,b.date)>=0 and
1719 intck('day',a.trading_date_plus2,b.date)<=0
1720 group by a.gvkey, a.fyearqtr;
1721 quit;
1722
1723 proc sql; create table out_raven1
1724 as select distinct a.*, mean(b.retail_volume/(1000*b.shrout)) as
1725 turn_retail_firm_0_1
1726 from out_raven1 as a left join trades_2003to2017 (where=(shrout>0)) as b
1727 on a.permno=b.permno and intck('day',a.trading_date,b.date)>=0 and
1728 intck('day',a.trading_date_plus1,b.date)<=0
1729 group by a.gvkey, a.fyearqtr;
1730 quit;
1731
1732 proc sql; create table out_raven1
1733 as select distinct a.*, mean(b.retail_volume/(1000*b.shrout)) as
1734 turn_retail_firm_0_3
1735 from out_raven1 as a left join trades_2003to2017 (where=(shrout>0)) as b
1736 on a.permno=b.permno and intck('day',a.trading_date,b.date)>=0 and
1737 intck('day',a.trading_date_plus3,b.date)<=0
1738 group by a.gvkey, a.fyearqtr;
1739 quit;
1740
1741 proc sql; create table out_raven1
1742 as select distinct a.*, mean(b.retail_volume/(1000*b.shrout)) as
1743 turn_retail_firm_0_4
1744 from out_raven1 as a left join trades_2003to2017 (where=(shrout>0)) as b
1745 on a.permno=b.permno and intck('day',a.trading_date,b.date)>=0 and
1746 intck('day',a.trading_date_plus4,b.date)<=0
1747 group by a.gvkey, a.fyearqtr;
1748 quit;
1749
1750 proc sql; create table out_raven1
1751 as select distinct a.*, mean(b.retail_volume/(1000*b.shrout)) as
1752 turn_retail_firm_0_5
1753 from out_raven1 as a left join trades_2003to2017 (where=(shrout>0)) as b
1754 on a.permno=b.permno and intck('day',a.trading_date,b.date)>=0 and
1755 intck('day',a.trading_date_plus5,b.date)<=0
1756 group by a.gvkey, a.fyearqtr;
1757 quit;
1758
1759 proc sql; create table out_raven1
1760 as select distinct a.*, mean(b.retail_volume/(1000*b.shrout)) as
1761 turnavg_retail_firm
1762 from out_raven1 as a left join trades_2003to2017 (where=(shrout>0)) as b
1763 on a.permno=b.permno and intck('day',a.trading_date_min41,b.date)>=0 and
1764 intck('day',a.trading_date_min11,b.date)<=0
1765 group by a.gvkey, a.fyearqtr;
1766 quit;
1767
1768 ** market retail volume;

```

```

1769
1770 proc sql; create table mkt_retail_turn as select distinct
1771 date, sum(retail_volume) as dailyretailvol, sum(buy_volume) as dailybuyvol,
1772 sum(sell_volume) as dailysellvol, sum(shrout) as dailytotshrout
1773 from trades_2003to2017 (where=(shrout>=0)) group by date;
1774 quit;
1775
1776 proc sql; create table out_raven1
1777 as select distinct a.*, mean(b.dailyretailvol/(1000*b.dailytotshrout)) as
1778 turn_event_mkt, mean(b.dailybuyvol/(1000*b.dailytotshrout)) as turn_buy_mkt,
1779 mean(b.dailysellvol/(1000*b.dailytotshrout)) as turn_sell_mkt
1780 from out_raven1 as a left join mkt_retail_turn as b
1781 on intck('day',a.trading_date,b.date)>=0 and
1782 intck('day',a.trading_date_plus2,b.date)<=0
1783 group by a.gvkey, a.fyearqtr;
1784 quit;
1785
1786 data out_raven1;
1787 set out_raven1;
1788 abn_turn_retail = turn_retail_firm - turntavg_retail_firm;
1789 abn_turn_retail_0_1 = turn_retail_firm_0_1 - turntavg_retail_firm;
1790 abn_turn_retail_0_3 = turn_retail_firm_0_3 - turntavg_retail_firm;
1791 abn_turn_retail_0_4 = turn_retail_firm_0_4 - turntavg_retail_firm;
1792 abn_turn_retail_0_5 = turn_retail_firm_0_5 - turntavg_retail_firm;
1793 drop turn_retail_firm: turntavg_retail_firm turn_event_mkt;
1794 run;
1795
1796 ** get buy and sell volume;
1797
1798 proc sql; create table out_raven1
1799 as select distinct a.*, mean(b.buy_volume/(1000*b.shrout)) as turn_buy_firm,
1800 mean(b.sell_volume/(1000*b.shrout)) as turn_sell_firm
1801 from out_raven1 as a left join trades_2003to2017 (where=(shrout>0)) as b
1802 on a.permno=b.permno and intck('day',a.trading_date,b.date)>=0 and
1803 intck('day',a.trading_date_plus2,b.date)<=0
1804 group by a.gvkey, a.fyearqtr;
1805 quit;
1806
1807 proc sql; create table out_raven1
1808 as select distinct a.*, mean(b.buy_volume/(1000*b.shrout)) as turntavg_buy_firm,
1809 mean(b.sell_volume/(1000*b.shrout)) as turntavg_sell_firm
1810 from out_raven1 as a left join trades_2003to2017 (where=(shrout>0)) as b
1811 on a.permno=b.permno and intck('day',a.trading_date_min41,b.date)>=0 and
1812 intck('day',a.trading_date_min11,b.date)<=0
1813 group by a.gvkey, a.fyearqtr;
1814 quit;
1815
1816 proc sql; create table out_raven1
1817 as select distinct a.*, mean(b.dailybuyvol/(1000*b.dailytotshrout)) as
1818 turntavg_buy_mkt, mean(b.dailysellvol/(1000*b.dailytotshrout)) as turntavg_sell_mkt
1819 from out_raven1 as a left join mkt_retail_turn as b
1820 on intck('day',a.trading_date_min41,b.date)>=0 and
1821 intck('day',a.trading_date_min11,b.date)<=0
1822 group by a.gvkey, a.fyearqtr;
1823 quit;
1824
1825 data out_raven2;
1826 set out_raven1;
1827 * abn_buy_event and abn_sell_event are mean firm and mean mkt adjusted;
1828 abn_buy_event = 100*(turn_buy_firm - turntavg_buy_firm - (turn_buy_mkt -
1829 turntavg_buy_mkt));

```

```

1830 abn_sell_event = 100*(turn_sell_firm - turntavg_sell_firm - (turn_sell_mkt -
1831 turntavg_sell_mkt));
1832 drop trading_date: turn_buy_mkt turn_sell_mkt turntavg_buy_mkt turntavg_sell_mkt
1833 turn_buy_firm turn_sell_firm turntavg_buy_firm turntavg_sell_firm;
1834 run;
1835
1836 %mend volume;
1837
1838 *****;
1839 ** add instown;
1840
1841 %macro instown;
1842
1843 ** add in WhaleWisdom institutional ownership;
1844
1845 proc sql;
1846     create table out_raven3
1847     as select distinct a.*, b.sum_shares as sum_shares_WW, b.shrout as shrout_WW,
1848     b.end_of_quarter_date
1849     from out_raven2 as a left join ww.matched_IO_2001_to_2016 as b
1850     on a.permno=b.permno
1851     and intnx('day',a.rdq,-100)<=b.end_of_quarter_date<=a.rdq
1852     order by a.gvkey, a.permno, b.end_of_quarter_date desc;quit;
1853
1854 proc sort data=out_raven3 nodupkey;
1855     by gvkey fyearqtr; quit;
1856
1857 data out_raven4;
1858     set out_raven3;
1859     inst_hold_WW = sum_shares_WW/(1000*shrout_WW);
1860     if inst_hold_WW = . then inst_hold_WW = 0;
1861     if inst_hold_WW > 1 then inst_hold_WW = 1;
1862     drop end_of_quarter_date sum_shares_WW shrout_WW;
1863 run;
1864
1865 proc sql;
1866     create table out_raven4
1867     as select distinct a.*, b.trading_date as rdq_trading
1868     from out_raven4 as a left join sas.trading_dates_final as b
1869     on a.rdq=b.orig_date;quit;
1870
1871 %mend instown;
1872
1873
1874 ***** export;
1875
1876 %macro export(dset, exportname);
1877
1878 proc export data=&dset
1879     outfile="&exportname."
1880     dbms=csv
1881     replace;
1882 run;
1883
1884 %mend export;
1885
1886 *****;
1887 *****;
1888 ** run the code here;
1889 *****;
1890 *****;

```

```

1891
1892  ** basic sample selection;
1893  %basic_info
1894
1895  ** zacks and ibes variables;
1896  %zacks_ibes
1897
1898  ** correct announcement times;
1899
1900  %anntimes
1901
1902  ** final sample before requiring UE/depth/volume vars;
1903
1904  %morevars
1905
1906  ** spreads and depths;
1907
1908  %spreadsdepths
1909
1910  ** more sample selection through the descriptives steps;
1911
1912  %descriptives
1913
1914  ** Ravenpack variables;
1915
1916  %raven
1917
1918
1919  ** Retail and MF volume;
1920
1921  %volume
1922
1923  ** Instown from WhaleWisdom;
1924
1925  %instown
1926
1927  ** export to csv;
1928
1929  %export(out_raven4, YOUR_DIR/Step1_final.csv)

```