

```

1 *****;
2 *      This file brings in the the main dataset from Step 1 and collects      *;
3 *      additional data (e.g., guidance, returns)                             *;
4 *      Last updated: 31 August 2018                                           *;
5 *****;
6
7 libname raw "YOUR_DIR";
8 libname home "YOUR_DIR";
9 libname master "YOUR_DIR";
10
11 * Bring in primary dataset from Step 1 *;
12 data main; set master.Step1_final; run;
13
14 * Pad gvkey for main dataset *;
15 data main; set main;
16     gvkey2 = put(gvkey,z6.);
17     drop gvkey;
18     rename gvkey2 = gvkey;
19
20     * This is for GVKEY 151928 *;
21     if permno = 89613 then permno = 16852;
22 run;
23
24 *****;
25 * Bring in the parsed AI article data for analyst presence *;
26 *****;
27
28 data AI_art; set master.AI_article_data; run;
29
30 * Merge *;
31 proc sql;
32     create table A1 as select
33         a.*, b.cons_earn_amt
34     from main as a left join AI_art as b
35     on a.gvkey = b.gvkey
36     and a.fyearqtr = b.fyearqtr;
37 quit;
38
39 * Let's now bring in various returns variables *;
40 * First calculate days since beginning of calendar year *;
41 data A2; set A1;
42     /* Note: use rdq_to_use variable here as this accounts for after-hours
43 releases */
44     if year(rdq_to_use) = 2012 then do;
45         ndays_cal = intnx('day',rdq_to_use,0) - intnx('day','01JAN2012'D,0);
46     end;
47     if year(rdq_to_use) = 2013 then do;
48         ndays_cal = intnx('day',rdq_to_use,0) - intnx('day','01JAN2013'D,0);
49     end;
50     if year(rdq_to_use) = 2014 then do;
51         ndays_cal = intnx('day',rdq_to_use,0) - intnx('day','01JAN2014'D,0);
52     end;
53     if year(rdq_to_use) = 2015 then do;
54         ndays_cal = intnx('day',rdq_to_use,0) - intnx('day','01JAN2015'D,0);
55     end;
56 run;
57
58 * Create variable for start date to accumulate TTM returns *;
59 data A3; set A2;
60     rdq_TTM = rdq_to_use - 365;
61     * Adjust for leap year *;

```

```

62         if rdq_to_use >= "28FEB2012"d and rdq_to_use <= "28FEB2013"d then do;
63             rdq_TTM = rdq_TTM - 1;
64         end;
65
66         format rdq_TTM rdq_to_use mmddyy8.;
67 run;
68
69 * Now let's obtain returns data from CRSP that is consistent with what is in the AI
70 articles *;
71 * Only upload limited identifier data to preserve upload speed *;
72 data B1; set A3(keep=gvkey permno fyearqtr rdq_to_use rdq_trading rdq_TTM); run;
73 proc sort data=B1 nodupkey; by gvkey fyearqtr; run;
74
75 * Bring in returns and volume from CRSP *;
76 * The AI articles reference raw returns excluding dividends, so use variable RETX
77 *;
78 rsubmit;
79 proc upload data=B1; run;
80 * First pull raw data then collapse after market returns pulled in *;
81 proc sql; create table crsp1
82     as select
83         a.*, b.retx, b.date
84     from B1 as a left join crsp.dsfs as b
85     on a.permno = b.permno
86     and a.rdq_to_use > b.date
87     and a.rdq_TTM <= b.date
88     order by a.permno, a.fyearqtr, b.date;
89 quit;
90
91 * Collapse down into firm-quarter level *;
92 proc sql; create table crsp2
93     as select distinct
94         permno, fyearqtr,
95         exp(sum(log(1+retx)))-1 as retxttm,
96         count(retx) as n_ttm
97     from crsp1
98     group by permno, fyearqtr;
99 quit;
100
101 * Do the exact same for YTD returns *;
102 proc sql; create table crsp3
103     as select
104         a.*, b.retx, b.date
105     from B1 as a left join crsp.dsfs as b
106     on a.permno = b.permno
107     and year(a.rdq_to_use) eq year(b.date)
108     and a.rdq_to_use > b.date
109     order by a.permno, a.fyearqtr, b.date;
110 quit;
111
112 * Collapse down into firm-quarter level *;
113 proc sql; create table crsp4
114     as select distinct
115         permno, fyearqtr,
116         exp(sum(log(1+retx)))-1 as retxytd,
117         count(retx) as n_ytd
118     from crsp3
119     group by permno, fyearqtr;
120 quit;
121
122 * Merge in both collapsed YTD and TTM data *;

```

```

123 proc sql; create table merge1
124     as select
125         a.*, b.*
126     from B1 as a left join crsp2 as b
127     on a.permno eq b.permno
128     and a.fyearqtr eq b.fyearqtr;
129 quit;
130
131 proc sql; create table merge2
132     as select
133         a.*, b.*
134     from merge1 as a left join crsp4 as b
135     on a.permno eq b.permno
136     and a.fyearqtr eq b.fyearqtr;
137 quit;
138
139 * Bring back locally *;
140 proc download data=merge2 out=B2; run;
141 endrsubmit;
142
143 * Merge in returns and volume data to main dataset *;
144 proc sql;
145     create table B3 as select
146         a.*, b.*
147     from A3 as a left join B2 as b
148     on a.gvkey = b.gvkey
149     and a.fyearqtr = b.fyearqtr
150     order by gvkey, fyearqtr;
151 quit;
152
153
154 *****;
155 ***   Bring in management forecast data   ***;
156 *****;
157
158 * First bring in Zacks forecasts - this is a raw file directly from WRDS *;
159 data zacks; set raw.zacks_preann_raw(drop=number_of_est); run;
160
161 * For now, only bring in guidance that is within [-1,+2] days of RDQ *;
162 proc sql;
163     create table C1 as select
164         a.*, b.*
165     from B3 as a left join zacks as b
166     on a.ZID = b.zid
167     and intnx('day',a.rdq_to_use,-1) <= b.announcement_date <=
168 intnx('day',a.rdq_to_use,2)
169     order by gvkey, fyearqtr;
170 quit;
171
172 * For now, only look at obs with non-missing Zacks guidance *;
173 data C2; set C1(keep=gvkey datadate rdq_to_use fyearqtr AI_cov zid ticker
174 measure_type period_type reference_period
175 announcement_date pri_dilt low high mid guidance
176 consensus consensus_std surprise comment);
177     where not missing(reference_period);
178 run;
179
180 * Create indicators for different types of guidance *;
181 data C3; set C2;
182     epsA = 0;
183     if (measure_type = "E" and period_type = 1) then epsA = 1;

```

```

184         epsQ = 0;
185         if (measure_type = "E" and period_type = 2) then epsQ = 1;
186 run;
187
188 * Pull max values for each variable before collapsing into unique gvkey-fyearqtr
189 rows *;
190 proc sql;
191     create table C4 as select
192         *, max(epsQ) as d_epsQ, max(epsA) as d_epsA
193     from C3
194     group by gvkey, fyearqtr;
195 quit;
196
197 * Down into firm-quarter level *;
198 proc sort data=C4 out=C5 nodupkey; by gvkey fyearqtr; run;
199
200 * Now create a variable that combines periodicities - just do for earnings *;
201 data C6; set C5;
202     mf_earn_zacks = max(d_epsQ,d_epsA);
203 run;
204
205 * Merge back into the full dataset *;
206 proc sql;
207     create table C7 as select
208         a.*, b.mf_earn_zacks
209     from B3 as a left join C6 as b
210     on a.gvkey eq b.gvkey
211     and a.fyearqtr eq b.fyearqtr
212     order by a.gvkey, b.fyearqtr;
213 quit;
214
215 * Check duplicates *;
216 proc sort data=C7 out=C7 nodupkey; by gvkey fyearqtr; run;
217
218 *** Now do the same for IBES ***;
219 * Bring in management guidance from IBES *;
220 data ibes; set raw.ibes_guide; run;
221
222 * Need to link with datasets based on identifiers *;
223 rsubmit;
224 libname home '~';
225 %include '/wrds/ibes/samples/iclink.sas';
226 proc download data=home.iclink out=work.iclink; run;
227 endrsubmit;
228
229 * Match IBES tickers to permnos *;
230 proc sql;
231     create table D1 as select
232         a.*, b.permno, b.score
233     from ibes as a left join iclink (where=(score in(0,1,2))) as b
234     on a.ticker = b.ticker;
235 quit;
236
237 proc sort data=D1 out=D2; by ticker anndats; run;
238
239 * Create indicators for different types of guidance *;
240 data D3; set D2;
241     earnA = 0;
242     if (measure
243 in("EPS", "EBT", "EBS", "GPS", "NET", "NETPAR", "OPR", "OPRPAR", "PRE", "PREPAR") and
244 pdicity = "ANN") then earnA = 1;

```

```

245         earnQ = 0;
246         if (measure
247 in("EPS","EBT","EBS","GPS","NET","NETPAR","OPR","OPRPAR","PRE","PREPAR") and
248 pdicity in("QTR","SAN")) then earnQ = 1;
249 run;
250
251 * This code allows us to collapse and retain same information *;
252 proc sql;
253     create table D4 as select
254         *, max(earnQ) as d_earnQ, max(earnA) as d_earnA
255     from D3
256     group by ticker, anndats;
257 quit;
258
259 data D5; set D4(drop=earnA earnQ); run;
260
261 * Now let's merge into the main dataset *;
262 proc sql;
263     create table D6 as select
264         a.*, b.ticker, b.anndats, b.d_earnQ, b.d_earnA
265     from C7 as a left join D5 as b
266     on a.permno = b.permno
267     and intnx('day',a.rdq_to_use,-1) <= b.anndats <= intnx('day',a.rdq_to_use,2)
268     order by gvkey, fyearqtr;
269 quit;
270
271 * Remove duplicates *;
272 proc sort data=D6 out=D7 nodupkey; by gvkey fyearqtr; run;
273
274 * Create indicators regardless of period *;
275 data D8; set D7;
276     mf_earn_ibes = max(d_earnQ,d_earnA);
277     drop d_earnQ d_earnA;
278 run;
279
280 * Reset missing values of guidance to zero *;
281 data D9; set D8;
282     if missing(mf_earn_zacks) then mf_earn_zacks = 0;
283     if missing(mf_earn_ibes) then mf_earn_ibes = 0;
284
285     * Create joint variables - whether guidance is in Zacks or IBES *;
286     guide_EA = max(mf_earn_zacks,mf_earn_ibes);
287 run;
288
289 * Save down for now *;
290 data master.Step2a_Final; set D9(drop=mf_earn_zacks mf_earn_ibes); run;
291
292
293 *****;
294 **           Bring in pre-EA returns and articles with intraday volume data           **;
295 *****;
296
297 data G1; set master.Step2a_Final; run;
298
299 * Merge in intraday data from CZ *;
300 data intra; set master.intraday_data; run;
301
302 * Need to pad gvkey for merge *;
303 data intra; set intra;
304     gvkey2 = put(gvkey,z6.);
305     drop gvkey;

```

```

306         rename gvkey2 = gvkey;
307 run;
308
309 * Merge *;
310 proc sql;
311     create table G1a as select
312         a.*, b.*
313     from G1 as a left join intra as b
314     on a.gvkey = b.gvkey
315     and a.fyearqtr = b.fyearqtr;
316 quit;
317
318 * Fix issue with one GVKEY-PERMNO merge *;
319 data G1b; set G1a;
320     * This is for GVKEY 151928 *;
321     if permno = 89613 then permno = 16852;
322 run;
323
324 * Subset data for ease of computation and create trading date ranges *;
325 data G2; set G1b; keep permno rdq_to_use; run;
326 proc sql;
327     create table G3 as select
328         a.*, b.trading_date_min5, b.trading_date_min1
329     from G2 as a left join master.trading_dates_final as b
330     on a.rdq_to_use = b.orig_date;
331 quit;
332
333 * Bring in pre-earnings announcement returns from CRSP *;
334 rsubmit;
335 proc upload data=G3; run;
336
337 * Do one week prior momentum [-5,-1] *;
338 proc sql;
339     create table newcrsp as select distinct
340         a.*, exp(sum(log(1+b.ret)))-1 as ret_pre_1week
341     from G3 as a left join crsp.dsff as b
342     on a.permno = b.permno
343     and a.trading_date_min5 <= b.date <= a.trading_date_min1
344     group by a.permno, a.rdq_to_use;
345 quit;
346
347 proc download data=newcrsp out=G4; run;
348 endrsubmit;
349
350 * Clean up dataset by removing trading date variables *;
351 data G5; set G4(drop=trading_date_min5 trading_date_min1); run;
352
353 * Merge back into main dataset *;
354 proc sql;
355     create table G6 as select
356         a.*, b.*
357     from G1b as a left join G5 as b
358     on a.permno = b.permno
359     and a.rdq_to_use = b.rdq_to_use;
360 quit;
361
362 * Convert intraday data to numeric *;
363 data G7; set G6;
364     array vars{*} before10 before5 before2 after10 after5 after2 diff10 diff5
365     diff2;

```

```

366         array varsa{*} before10a before5a before2a after10a after5a after2a diff10a
367 diff5a diff2a;
368         do i = 1 to dim(vars);
369             varsa{i} = input(vars{i},best12.);
370         drop i;
371     end;
372
373     drop before10 before5 before2 after10 after5 after2 diff10 diff5 diff2;
374 run;
375
376 * Save down *;
377 data master.Step2a_Final; set G7; run;
378
379
380 *****;
381 * Go back and grab guidance data in period before EA *;
382 *****;
383
384 * Clean up missing tickers within group *;
385 proc sort data=G7 out=M1; by gvkey fyearqtr descending ticker; run;
386 proc sql;
387     create table M1a as select
388         gvkey, max(ticker) as ticker
389     from M1
390     group by gvkey;
391 quit;
392 proc sql;
393     create table M1b as select
394         a.*, b.ticker
395     from M1(drop=ticker) as a left join M1a as b
396     on a.gvkey = b.gvkey
397     order by gvkey, fyearqtr;
398 quit;
399 * Create lagged RDQ - if missing, then set to 55 days prior *;
400 data M1c; set M1b;
401     by gvkey;
402     lag_rdq_to_use = lag(rdq_to_use);
403     if first.gvkey then lag_rdq_to_use = .;
404
405     * Reset if missing or if beyond 90 days *;
406     * 55 days is conservative to not pick up forecasts *;
407     if lag_rdq_to_use = . then lag_rdq_to_use = intnx('day',rdq_to_use,-55);
408     if intck('day',lag_rdq_to_use,rdq_to_use) > 90 then lag_rdq_to_use =
409 intnx('day',rdq_to_use,-55);
410
411     format lag_rdq_to_use MMDDYY8.;
412 run;
413
414 * Bring in Zacks forecast data *;
415 * From 2 days prior quarter EA to 2 days before current *;
416 data zacks; set raw.zacks_preann_raw(drop=number_of_est); run;
417 proc sql;
418     create table I1 as select
419         a.*, b.*, (b.announcement_date - a.rdq_to_use) as guide_lapse
420     from M1C as a left join zacks as b
421     on a.ZID = b.zid
422     and intnx('day',a.lag_rdq_to_use,2) <= b.announcement_date <=
423 intnx('day',a.rdq_to_use,-2)
424     order by gvkey, fyearqtr;
425 quit;
426

```

```

427 * For now, only look at obs with non-missing Zacks guidance *;
428 data I2; set I1;
429     where not missing(reference_period);
430 run;
431
432 * Create indicators for different types of guidance *;
433 data I3; set I2;
434     epsA = 0;
435     if (measure_type = "E" and period_type = 1) then epsA = 1;
436     epsQ = 0;
437     if (measure_type = "E" and period_type = 2) then epsQ = 1;
438
439     * Negative earnings guidance *;
440     epsA_neg = 0;
441     if (measure_type = "E" and period_type = 1 and (mid - consensus) < 0) then
442 epsA_neg = 1;
443     epsQ_neg = 0;
444     if (measure_type = "E" and period_type = 2 and (mid - consensus) < 0) then
445 epsQ_neg = 1;
446
447     * Date variable if quarterly guidance *;
448     lapse_earn = .;
449     if epsQ = 1 then lapse_earn = guide_lapse;
450     * And if negative guidance *;
451     lapse_earn_neg = .;
452     if epsQ_neg then lapse_earn_neg = guide_lapse;
453 run;
454
455 * Pull max values for each variable before collapsing into unique gvkey-fyearqtr
456 rows *;
457 proc sql;
458     create table I4 as select
459         *, max(epsQ) as d_epsQ, max(epsA) as d_epsA,
460         max(epsQ_neg) as d_epsQneg, max(epsA_neg) as d_epsAneg, max(lapse_earn)
461 as lapse, max(lapse_earn_neg) as lapse_neg
462     from I3
463     group by gvkey, fyearqtr;
464 quit;
465
466 * Down into firm-quarter level *;
467 proc sort data=I4 out=I5 nodupkey; by gvkey fyearqtr; run;
468
469 * Now create a variable that combines periodicities - just do for earnings *;
470 data I6; set I5;
471     mf_earn_zacks = max(d_epsQ,d_epsA);
472     * Keep just quarterly variable too *;
473     mf_earnQ_zacks = d_epsQ;
474
475     * Keep separate for negative warnings - only consider quarterly *;
476     warn_earn_zacks = d_epsQneg;
477 run;
478
479 * Merge back into the full dataset *;
480 proc sql;
481     create table I7 as select
482         a.*, b.mf_earn_zacks, b.mf_earnQ_zacks, b.warn_earn_zacks, b.lapse,
483 b.lapse_neg
484     from M1c as a left join I6 as b
485     on a.gvkey eq b.gvkey
486     and a.fyearqtr eq b.fyearqtr
487     order by a.gvkey, b.fyearqtr;

```



```

488 quit;
489
490 * Bring in guidance data - repull to grab for 2011Q4 *;
491 * Bring in management guidance from IBES *;
492 data raw.ibes_guide2; set ibes; run;
493
494 data J1; set raw.ibes_guide2; run;
495 * Fix up dates for merging with fyearqtr *;
496 data J2; set J1;
497     length fyearqtr 5.;
498     month = 0;
499     if prd_mon in(1,2,3) then month = 1;
500     if prd_mon in(4,5,6) then month = 2;
501     if prd_mon in(7,8,9) then month = 3;
502     if prd_mon in(10,11,12) then month = 4;
503
504     year2 = put(prd_yr,4.);
505     month2 = put(month,1.);
506
507     fyearqtr2 = cats(' ',year2,month2);
508     fyearqtr = input(fyearqtr2,5.);
509
510     drop month year2 month2 fyearqtr2;
511 run;
512 proc sort data=J2 out=J2a; by ticker anndats; run;
513
514 * Create indicators for different types of guidance *;
515 data J3; set J2a;
516     earnA = 0;
517     if (measure
518 in("EPS","EBT","EBS","GPS","NET","NETPAR","OPR","OPRPAR","PRE","PREPAR") and
519 pdicity = "ANN") then earnA = 1;
520     earnQ = 0;
521     if (measure
522 in("EPS","EBT","EBS","GPS","NET","NETPAR","OPR","OPRPAR","PRE","PREPAR") and
523 pdicity in("QTR","SAN")) then earnQ = 1;
524
525     * Negative quarterly guidance *;
526     earnQ_neg = 0;
527     if earnQ = 1 and guidance_code in("01","06") then earnQ_neg = 1;
528 run;
529
530 * This code allows us to collapse and retain same information *;
531 proc sql;
532     create table J4 as select
533         *, max(earnQ) as d_earnQ, max(earnA) as d_earnA, max(earnQ_neg) as
534 d_earnQneg
535     from J3
536     group by ticker, anndats;
537 quit;
538 data J5; set J4(drop=earnA earnQ earnQ_neg); run;
539
540 * Now let's merge into the main dataset *;
541 * Same as for the Zacks data *;
542 proc sort data=I7; by gvkey fyearqtr; run;
543 proc sql;
544     create table J6 as select
545         a.*, b.anndats, b.d_earnQ, b.d_earnA, b.d_earnQneg,
546         (b.anndats - a.rdq_to_use) as guide_lapse
547     from I7 as a left join J5 as b
548     on a.ticker = b.ticker

```

```

549         and intnx('day',a.lag_rdq_to_use,2) <= b.anndats <=
550 intnx('day',a.rdq_to_use,-2)
551     order by gvkey, fyearqtr;
552 quit;
553 * Adjust lapse variable to only be for quarterly earnings *;
554 data J6a; set J6;
555     i_lapse = .;
556     if d_earnQ = 1 then i_lapse = guide_lapse;
557     i_lapse_neg = .;
558     if d_earnQneg = 1 then i_lapse_neg = guide_lapse;
559     drop guide_lapse;
560 run;
561 * Remove duplicates *;
562 proc sort data=J6a out=J7 nodupkey; by gvkey fyearqtr; run;
563
564 * Create indicators regardless of period *;
565 data J8; set J7;
566     mf_earn_ibes = max(d_earnQ,d_earnA);
567     * Just quarterly earnings *;
568     mf_earnQ_ibes = d_earnQ;
569
570     * Negative earnings warning *;
571     warn_earn_ibes = d_earnQneg;
572
573     drop d_earnQ d_earnA d_earnQneg;
574 run;
575
576 * Reset missing values of guidance to zero *;
577 data J9; set J8;
578     if missing(mf_earn_zacks) then mf_earn_zacks = 0;
579     if missing(mf_earnQ_zacks) then mf_earnQ_zacks = 0;
580     if missing(warn_earn_zacks) then warn_earn_zacks = 0;
581     if missing(mf_earn_ibes) then mf_earn_ibes = 0;
582     if missing(mf_earnQ_ibes) then mf_earnQ_ibes = 0;
583     if missing(warn_earn_ibes) then warn_earn_ibes = 0;
584
585     * Create joint variables - whether guidance is in Zacks or IBES *;
586     pre_guide_earn = max(mf_earn_zacks,mf_earn_ibes);
587     pre_guide_earnQ = max(mf_earnQ_zacks,mf_earnQ_ibes);
588     warn_earn = max(warn_earn_ibes,warn_earn_zacks);
589
590     * Combine date lapse variables *;
591     preempt_time = max(lapse,i_lapse);
592     preempt_time_neg = max(lapse_neg,i_lapse_neg);
593
594     drop mf_earn_zacks mf_earn_ibes warn_earn_zacks warn_earn_ibes mf_earnQ_zacks
595 mf_earnQ_ibes;
596     drop lapse lapse_neg i_lapse i_lapse_neg;
597 run;
598
599 * Save down *;
600 data master.Step2_Final; set J9; run;
601
602
603 *****;
604 **     END OF FILE - OVER TO STATA FOR ADDITIONAL MERGING
605 **     AND VARIABLE CREATION
606 *****;
607
608

```