

```

1 *****;
2 *** Code to pull and calculated intraday realized volatility (TAQ) ***;
3 *** Last updated: 31 August 2018 ***;
4 *****;
5
6 libname temp 'YOUR_DIR';
7 libname taqms 'YOUR_DIR';
8 options errors=50;
9 options msglevel=i fullstimer;
10
11 /*****
12 /* Pull TAQ Master Files and CRSP Stocknames to create linking table */
13 /*****
14
15 /*Code to pull the TAQ master files*/
16
17 %macro pull_master(startyear=,endyear=,startmonth=,endmonth=,startday=,endday=);
18
19 /*Create empty output datasets to append to later*/
20
21 DATA temp.master;
22     LENGTH DATE 4 LISTED_MARKET $ 1 TAPE $ 1 UOT 8 CUSIP $ 9 SYMBOL $ 16 ;
23     FORMAT DATE DATE9.;
24     INFORMAT DATE DATE9.;
25     STOP;
26 RUN;
27
28 %do y=&startyear %to &endyear; /*cycling through years*/
29     %do m=&startmonth %to &endmonth; /*cycling through months*/
30         %let prefixm=0;
31         %if &m>=10 %then %let prefixm=;
32             %do d=&startday %to &endday; /*cycling through days*/
33                 %let prefixd=0;
34                 %if &m>=10 %then %let prefixd=;
35                 %if %sysfunc(exist(taqms.mastm_&y&prefixm&m&prefixd&d)) %then
36 %do; /*checking to make sure it exists*/
37                     data temptaq;
38                         set taqms.mastm_&y&prefixm&m&prefixd&d;
39                         symbol=cats(symbol_root,symbol_suffix);
40                         drop symbol_root symbol_suffix symbol_15 cqs_symbol
41 sec_desc bbg_bsid bbg_gid tron_amex tron_bx tron_nsx tron_finra tron_ise
42 tron_edge_a tron_edge_x tron_chx tron_nyse tron_arca tron_nasdaq tron_cts tron_cboe
43 tron_psx tron_batsy tron_bats; run;
44                     proc append base=temp.master data=temptaq force; run;
45                     %end;
46                 %end;
47             %end;
48         %end;
49     %end;
50 %mend pull_master;
51
52 ** This is where you actually run the code;
53 %pull_master(startyear=2011,endyear=2015,startmonth=1,endmonth=12,startday=1,endday
54 =31);
55
56
57
58 *****;
59 ** Code to calculate realized intraday volatility **;
60 ** Last updated: February 28, 2018 **;
61 *****;

```

```

62
63 * NOTE - DUE TO PROCESSING TIME, DO EACH YEAR SEPARATELY *;
64 * RESULTS IN SEVERAL DIFFERENT DATASETS WHICH ARE STACKED TOGETHER IN THE END *;
65
66
67 * Library preamble *;
68 libname taq 'YOUR_DIR';
69 libname home "YOUR_DIR";
70
71 * Input data *;
72 %let taq_ds=taq.ctm_2011;; * dataset(s) of interest;
73 %let start_time = '9:30:00't; * starting time;
74 %let interval_seconds = 5*60; * interval is 5*60 seconds, 5 min;
75 %let n_intervals = (23400/60/5)+1; * number of intervals for a given day
76
77 * Extract data available;
78 * Time is between 9:30am and 4:01pm (extend one minute to pick up final trades);
79 data A1; set &taq_ds(keep=sym_root sym_suffix price Tr_corr time_m date);
80     where time_m between '9:30:00't and '16:01:00't
81
82     /* Per Holden and Jacobsen, include this code for the trades file */
83     and Tr_corr eq '00' and price gt 0;
84
85     /* Adjust symbol to account for different share classes */
86     length symbol $16;
87     symbol = cats(sym_root,sym_suffix);
88     keep symbol date price time_m;
89 run;
90
91 proc sort data=A1; by symbol date time_m; run;
92
93 data A2; set A1;
94     by symbol date time_m;
95     format itime rtime time12.;
96     if first.symbol=1 or first.date=1 then do;
97         *Initialize time and price when new symbol or date starts;
98         rtime=time_m;
99         iprice=price;
100        itime= &start_time;
101    end;
102    if time_m >= itime then do; *Interval reached;
103        output; *rtime and iprice hold the last observation values;
104        itime = itime + &interval_seconds;
105        do while(time_m >= itime); *need to fill in all time intervals;
106            output;
107            itime = itime + &interval_seconds;
108        end;
109    end;
110    rtime=time_m;
111    iprice=price;
112    retain itime rtime iprice; *Carry time and price values forward;
113 run;
114
115 * Calculate log returns and squared log returns *;
116 * Reset odd trades that result in >= 50% returns *;
117 data A3; set A2;
118     by symbol date;
119     if first.symbol=1 or first.date=1 then do;
120         *Initialize time and price when new symbol or date starts;
121         group=1;
122         end;

```

```

123     else do;
124         group=group+1;
125     end;
126     if group=&n_intervals then group=0;
127         return=log(iprice/lag(iprice));
128         if abs(return)>=.5 then return=0;
129         retsq=return*return;
130         if symbol ^= lag(symbol) then return=0;
131         if symbol ^= lag(symbol) then retsq=0;
132     retain group;
133 run;
134
135 * Computes realized volatility on a daily basis using 5-minute intervals *;
136 proc sql;
137     create table RVday as
138     select a.symbol,a.date,sum(a.return) as SumRet, sum(a.retsq) as RVday
139     from A3 as a
140     group by a.date, a.symbol;
141 quit;
142
143 * Save down dataset *;
144 data home.RVday_2011; set RVday; run;
145
146
147 *****;
148 **          SAME FOR 2012          **;
149 *****;
150
151
152
153 * Input data *;
154 %let taq_ds=taq.ctm_2012;; * dataset(s) of interest;
155 %let start_time = '9:30:00't; * starting time;
156 %let interval_seconds = 5*60; * interval is 5*60 seconds, 5 min;
157 %let n_intervals = (23400/60/5)+1; * number of intervals for a given day
158
159 * Extract data available;
160 * Time is between 9:30am and 4:01pm (extend one minute to pick up final trades);
161 data A1; set &taq_ds(keep=sym_root sym_suffix price Tr_corr time_m date);
162     where time_m between '9:30:00't and '16:01:00't
163
164     /* Per Holden and Jacobsen, include this code for the trades file */
165     and Tr_corr eq '00' and price gt 0;
166
167     /* Adjust symbol to account for different share classes */
168     length symbol $16;
169     symbol = cats(sym_root,sym_suffix);
170     keep symbol date price time_m;
171 run;
172
173 proc sort data=A1; by symbol date time_m; run;
174
175 data A2; set A1;
176     by symbol date time_m;
177     format itime rtime time12.;
178     if first.symbol=1 or first.date=1 then do;
179         *Initialize time and price when new symbol or date starts;
180         rtime=time_m;
181         iprice=price;
182         itime= &start_time;
183     end;

```

```

184         if time_m >= itime then do; *Interval reached;
185             output; *rtime and iprice hold the last observation values;
186             itime = itime + &interval_seconds;
187             do while(time_m >= itime); *need to fill in all time intervals;
188                 output;
189                 itime = itime + &interval_seconds;
190             end;
191         end;
192         rtime=time_m;
193         iprice=price;
194         retain itime rtime iprice; *Carry time and price values forward;
195 run;
196
197 * Calculate log returns and squared log returns *;
198 * Reset odd trades that result in >= 50% returns *;
199 data A3; set A2;
200     by symbol date;
201     if first.symbol=1 or first.date=1 then do;
202         *Initialize time and price when new symbol or date starts;
203         group=1;
204         end;
205     else do;
206         group=group+1;
207     end;
208     if group=&n_intervals then group=0;
209     return=log(iprice/lag(iprice));
210     if abs(return)>=0.5 then return=0;
211     retsq=return*return;
212     if symbol ^= lag(symbol) then return=0;
213     if symbol ^= lag(symbol) then retsq=0;
214     retain group;
215 run;
216
217 * Computes realized volatility on a daily basis using 5-minute intervals *;
218 proc sql;
219     create table RVday as
220     select a.symbol,a.date,sum(a.return) as SumRet, sum(a.retsq) as RVday
221     from A3 as a
222     group by a.date, a.symbol;
223 quit;
224
225 * Save down dataset *;
226 data home.RVday_2012; set RVday; run;
227
228
229
230 *****;
231 **          SAME FOR 2013          **;
232 *****;
233
234 * Input data *;
235 %let taq_ds=taq.ctm_2013;; * dataset(s) of interest;
236 %let start_time = '9:30:00't; * starting time;
237 %let interval_seconds = 5*60; * interval is 5*60 seconds, 5 min;
238 %let n_intervals = (23400/60/5)+1; * number of intervals for a given day
239
240 * Extract data available;
241 * Time is between 9:30am and 4:01pm (extend one minute to pick up final trades);
242 data A1; set &taq_ds(keep=sym_root sym_suffix price Tr_corr time_m date);
243     where time_m between '9:30:00't and '16:01:00't
244

```

```

245      /* Per Holden and Jacobsen, include this code for the trades file */
246      and Tr_corr eq '00' and price gt 0;
247
248      /* Adjust symbol to account for different share classes */
249      length symbol $16;
250      symbol = cats(sym_root,sym_suffix);
251      keep symbol date price time_m;
252  run;
253
254  proc sort data=A1; by symbol date time_m; run;
255
256  data A2; set A1;
257      by symbol date time_m;
258      format itime rtime time12.;
259      if first.symbol=1 or first.date=1 then do;
260          *Initialize time and price when new symbol or date starts;
261          rtime=time_m;
262          iprice=price;
263          itime= &start_time;
264          end;
265          if time_m >= itime then do; *Interval reached;
266              output; *itime and iprice hold the last observation values;
267              itime = itime + &interval_seconds;
268              do while(time_m >= itime); *need to fill in all time intervals;
269                  output;
270                  itime = itime + &interval_seconds;
271              end;
272          end;
273          rtime=time_m;
274          iprice=price;
275          retain itime rtime iprice; *Carry time and price values forward;
276  run;
277
278  * Calculate log returns and squared log returns *;
279  * Reset odd trades that result in >= 50% returns *;
280  data A3; set A2;
281      by symbol date;
282      if first.symbol=1 or first.date=1 then do;
283          *Initialize time and price when new symbol or date starts;
284          group=1;
285          end;
286      else do;
287          group=group+1;
288      end;
289      if group=&n_intervals then group=0;
290      return=log(iprice/lag(iprice));
291      if abs(return)>=.5 then return=0;
292      retsq=return*return;
293      if symbol ^= lag(symbol) then return=0;
294      if symbol ^= lag(symbol) then retsq=0;
295      retain group;
296  run;
297
298  * Computes realized volatility on a daily basis using 5-minute intervals *;
299  proc sql;
300      create table RVday as
301      select a.symbol,a.date,sum(a.return) as SumRet, sum(a.retsq) as RVday
302      from A3 as a
303      group by a.date, a.symbol;
304  quit;
305

```

```

306 * Save down dataset *;
307 data home.RVday_2013; set RVday; run;
308
309
310
311 *****;
312 ** SAME FOR 2014 - PT1 **;
313 *****;
314
315 * Input data *;
316 %let taq_ds=taq.ctm_201401: taq.ctm_201402: taq.ctm_201403:
317 taq.ctm_201404: taq.ctm_201405: taq.ctm_201406;; * dataset(s) of
318 interest;
319 %let start_time = '9:30:00't; * starting time;
320 %let interval_seconds = 5*60; * interval is 5*60 seconds, 5 min;
321 %let n_intervals = (23400/60/5)+1; * number of intervals for a given day
322
323 * Extract data available;
324 * Time is between 9:30am and 4:01pm (extend one minute to pick up final trades);
325 data A1; set &taq_ds(keep=sym_root sym_suffix price Tr_corr time_m date);
326 where time_m between '9:30:00't and '16:01:00't
327
328 /* Per Holden and Jacobsen, include this code for the trades file */
329 and Tr_corr eq '00' and price gt 0;
330
331 /* Adjust symbol to account for different share classes */
332 length symbol $16;
333 symbol = cats(sym_root,sym_suffix);
334 keep symbol date price time_m;
335 run;
336
337 proc sort data=A1; by symbol date time_m; run;
338
339 data A2; set A1;
340 by symbol date time_m;
341 format itime rtime time12.;
342 if first.symbol=1 or first.date=1 then do;
343 *Initialize time and price when new symbol or date starts;
344 rtime=time_m;
345 iprice=price;
346 itime= &start_time;
347 end;
348 if time_m >= itime then do; *Interval reached;
349 output; *rtime and iprice hold the last observation values;
350 itime = itime + &interval_seconds;
351 do while(time_m >= itime); *need to fill in all time intervals;
352 output;
353 itime = itime + &interval_seconds;
354 end;
355 end;
356 rtime=time_m;
357 iprice=price;
358 retain itime rtime iprice; *Carry time and price values forward;
359 run;
360
361 * Calculate log returns and squared log returns *;
362 * Reset odd trades that result in >= 50% returns *;
363 data A3; set A2;
364 by symbol date;
365 if first.symbol=1 or first.date=1 then do;
366 *Initialize time and price when new symbol or date starts;

```

```

367     group=1;
368     end;
369     else do;
370         group=group+1;
371     end;
372     if group=&n_intervals then group=0;
373     return=log(iprice/lag(iprice));
374     if abs(return)>=.5 then return=0;
375     retsq=return*return;
376     if symbol ^= lag(symbol) then return=0;
377     if symbol ^= lag(symbol) then retsq=0;
378     retain group;
379 run;
380
381 * Computes realized volatility on a daily basis using 5-minute intervals *;
382 proc sql;
383     create table RVday as
384     select a.symbol,a.date,sum(a.return) as SumRet, sum(a.retsq) as RVday
385     from A3 as a
386     group by a.date, a.symbol;
387 quit;
388
389 * Save down dataset *;
390 data home.RVday_2014_pt1; set RVday; run;
391
392 *****;
393 **     SAME FOR 2014 - PT2     **;
394 *****;
395
396
397 * Input data *;
398 %let taq_ds=taq.ctm_201407: taq.ctm_201408: taq.ctm_201409:
399     taq.ctm_201410: taq.ctm_201411: taq.ctm_201412;; * dataset(s) of
400 interest;
401 %let start_time = '9:30:00't; * starting time;
402 %let interval_seconds = 5*60; * interval is 5*60 seconds, 5 min;
403 %let n_intervals = (23400/60/5)+1; * number of intervals for a given day
404
405 * Extract data available;
406 * Time is between 9:30am and 4:01pm (extend one minute to pick up final trades);
407 data A1; set &taq_ds(keep=sym_root sym_suffix price Tr_corr time_m date);
408     where time_m between '9:30:00't and '16:01:00't
409
410     /* Per Holden and Jacobsen, include this code for the trades file */
411     and Tr_corr eq '00' and price gt 0;
412
413     /* Adjust symbol to account for different share classes */
414     length symbol $16;
415     symbol = cats(sym_root,sym_suffix);
416     keep symbol date price time_m;
417 run;
418
419 proc sort data=A1; by symbol date time_m; run;
420
421 data A2; set A1;
422     by symbol date time_m;
423     format itime rtime time12.;
424     if first.symbol=1 or first.date=1 then do;
425         *Initialize time and price when new symbol or date starts;
426         rtime=time_m;
427         iprice=price;

```

```

428         itime= &start_time;
429     end;
430     if time_m >= itime then do; *Interval reached;
431         output; *rtime and iprice hold the last observation values;
432         itime = itime + &interval_seconds;
433         do while(time_m >= itime); *need to fill in all time intervals;
434             output;
435             itime = itime + &interval_seconds;
436         end;
437     end;
438     rtime=time_m;
439     iprice=price;
440     retain itime rtime iprice; *Carry time and price values forward;
441 run;
442
443 * Calculate log returns and squared log returns *;
444 * Reset odd trades that result in >= 50% returns *;
445 data A3; set A2;
446     by symbol date;
447     if first.symbol=1 or first.date=1 then do;
448         *Initialize time and price when new symbol or date starts;
449         group=1;
450     end;
451     else do;
452         group=group+1;
453     end;
454     if group=&n_intervals then group=0;
455     return=log(iprice/lag(iprice));
456     if abs(return)>=.5 then return=0;
457     retsq=return*return;
458     if symbol ^= lag(symbol) then return=0;
459     if symbol ^= lag(symbol) then retsq=0;
460     retain group;
461 run;
462
463 * Computes realized volatility on a daily basis using 5-minute intervals *;
464 proc sql;
465     create table RVday as
466     select a.symbol,a.date,sum(a.return) as SumRet, sum(a.retsq) as RVday
467     from A3 as a
468     group by a.date, a.symbol;
469 quit;
470
471 * Save down dataset *;
472 data home.RVday_2014_pt2; set RVday; run;
473
474
475
476 *****;
477 **     SAME FOR 2015 - PT1     **;
478 *****;
479
480 * Input data *;
481 %let taq_ds=taq.ctm_201501: taq.ctm_201502: taq.ctm_201503;; * dataset(s) of
482 interest;
483 %let start_time = '9:30:00't; * starting time;
484 %let interval_seconds = 5*60; * interval is 5*60 seconds, 5 min;
485 %let n_intervals = (23400/60/5)+1; * number of intervals for a given day
486
487 * Extract data available;
488 * Time is between 9:30am and 4:01pm (extend one minute to pick up final trades);

```



```

489 data A1; set &taq_ds(keep=sym_root sym_suffix price Tr_corr time_m date);
490     where time_m between '9:30:00't and '16:01:00't
491
492     /* Per Holden and Jacobsen, include this code for the trades file */
493     and Tr_corr eq '00' and price gt 0;
494
495     /* Adjust symbol to account for different share classes */
496     length symbol $16;
497     symbol = cats(sym_root,sym_suffix);
498     keep symbol date price time_m;
499 run;
500
501 proc sort data=A1; by symbol date time_m; run;
502
503 data A2; set A1;
504     by symbol date time_m;
505     format itime rtime time12.;
506     if first.symbol=1 or first.date=1 then do;
507         *Initialize time and price when new symbol or date starts;
508         rtime=time_m;
509         iprice=price;
510         itime= &start_time;
511     end;
512     if time_m >= itime then do; *Interval reached;
513         output; *rtime and iprice hold the last observation values;
514         itime = itime + &interval_seconds;
515         do while(time_m >= itime); *need to fill in all time intervals;
516             output;
517             itime = itime + &interval_seconds;
518         end;
519     end;
520     rtime=time_m;
521     iprice=price;
522     retain itime rtime iprice; *Carry time and price values forward;
523 run;
524
525 * Calculate log returns and squared log returns *;
526 * Reset odd trades that result in >= 50% returns *;
527 data A3; set A2;
528     by symbol date;
529     if first.symbol=1 or first.date=1 then do;
530         *Initialize time and price when new symbol or date starts;
531         group=1;
532     end;
533     else do;
534         group=group+1;
535     end;
536     if group=&n_intervals then group=0;
537     return=log(iprice/lag(iprice));
538     if abs(return)>=.5 then return=0;
539     retsq=return*return;
540     if symbol ^= lag(symbol) then return=0;
541     if symbol ^= lag(symbol) then retsq=0;
542     retain group;
543 run;
544
545 * Computes realized volatility on a daily basis using 5-minute intervals *;
546 proc sql;
547     create table RVday as
548     select a.symbol,a.date,sum(a.return) as SumRet, sum(a.retsq) as RVday
549     from A3 as a

```

```

550         group by a.date, a.symbol;
551 quit;
552
553 * Save down dataset *;
554 data home.RVday_2015_pt1; set RVday; run;
555
556
557
558 *****;
559 **      SAME FOR 2015 - PT2      **;
560 *****;
561
562 * Input data *;
563 %let taq_ds=taq.ctm_201504: taq.ctm_201505: taq.ctm_201506:
564         taq.ctm_201507: taq.ctm_201508: taq.ctm_201509:
565         taq.ctm_201510: taq.ctm_201511: taq.ctm_201512;; * dataset(s) of
566 interest;
567 %let start_time = '9:30:00't; * starting time;
568 %let interval_seconds = 5*60; * interval is 5*60 seconds, 5 min;
569 %let n_intervals = (23400/60/5)+1; * number of intervals for a given day
570
571 * Extract data available;
572 * Time is between 9:30am and 4:01pm (extend one minute to pick up final trades);
573 data A1; set &taq_ds(keep=sym_root sym_suffix price Tr_corr time_m date);
574         where time_m between '9:30:00't and '16:01:00't
575
576         /* Per Holden and Jacobsen, include this code for the trades file */
577         and Tr_corr eq '00' and price gt 0;
578
579         /* Adjust symbol to account for different share classes */
580         length symbol $16;
581         symbol = cats(sym_root,sym_suffix);
582         keep symbol date price time_m;
583 run;
584
585 proc sort data=A1; by symbol date time_m; run;
586
587 data A2; set A1;
588         by symbol date time_m;
589         format itime rtime time12.;
590         if first.symbol=1 or first.date=1 then do;
591             *Initialize time and price when new symbol or date starts;
592             rtime=time_m;
593             iprice=price;
594             itime= &start_time;
595             end;
596         if time_m >= itime then do; *Interval reached;
597             output; *rtime and iprice hold the last observation values;
598             itime = itime + &interval_seconds;
599             do while(time_m >= itime); *need to fill in all time intervals;
600                 output;
601                 itime = itime + &interval_seconds;
602             end;
603         end;
604         rtime=time_m;
605         iprice=price;
606         retain itime rtime iprice; *Carry time and price values forward;
607 run;
608
609 * Calculate log returns and squared log returns *;
610 * Reset odd trades that result in >= 50% returns *;

```

```

611 data A3; set A2;
612     by symbol date;
613     if first.symbol=1 or first.date=1 then do;
614         *Initialize time and price when new symbol or date starts;
615         group=1;
616     end;
617     else do;
618         group=group+1;
619     end;
620     if group=&n_intervals then group=0;
621     return=log(iprice/lag(iprice));
622     if abs(return)>=.5 then return=0;
623     retsq=return*return;
624     if symbol ^= lag(symbol) then return=0;
625     if symbol ^= lag(symbol) then retsq=0;
626     retain group;
627 run;
628
629 * Computes realized volatility on a daily basis using 5-minute intervals *;
630 proc sql;
631     create table RVday as
632     select a.symbol,a.date,sum(a.return) as SumRet, sum(a.retsq) as RVday
633     from A3 as a
634     group by a.date, a.symbol;
635 quit;
636
637 * Save down dataset *;
638 data home.RVday_2015_pt2; set RVday; run;
639
640
641
642
643 *****;
644 **             Match TAQ to Permno and bring in realized volatility
645 **;
646 *****;
647
648 libname taqmast 'YOUR_DIR';
649 libname home 'YOUR_DIR';
650
651 * Pull CUSIP and permno from CRSP quarterly stocknames file *;
652 proc sql;
653     create table stocknames
654     as select distinct permno, substr(ncusip,1,8) as ncusip, namedt format
655     date9.,
656         nameenddt format date9.
657     from crspq.stocknames
658     group by permno, ncusip;
659 quit;
660
661 **Adjusting master table from TAQ;
662 proc sql;
663     create table taqtempl
664     as select distinct symbol,
665         substr(cusip,1,8) as taq_cusip, date as table_date format date9.
666     from taqmast.master
667     order by symbol, calculated taq_cusip, table_date;
668 quit;
669
670 * Link datasets together *;
671 proc sql;

```

```

672         create table taqtemp2
673         as select distinct a.symbol, a.taq_cusip, b.permno, a.table_date
674         from taqtemp1 as a join stocknames as b
675             on a.taq_cusip = b.ncusip AND (b.namedt <= a.table_date OR b.namedt =
676         .B)
677             AND (a.table_date <= b.nameenddt OR b.nameenddt >= '30dec2016'd)
678         order by b.permno, a.table_date;
679 quit;
680
681 proc sort data=taqtemp2 nodupkey; by permno table_date; run;
682 proc sort data=taqtemp2 out=home.taq_match; by permno table_date; run;
683
684 *****;
685
686 * Next, stack data on realized volatility from TAQ into one dataset *;
687 data RVmain; set home.RVday_2011 home.RVday_2012 home.RVday_2013
688             home.RVday_2014_pt1 home.RVday_2014_pt2 home.RVday_2015_pt1
689             home.RVday_2015_pt2; run;
690
691 * Link up master dataset with RV dataset *;
692 proc sql;
693     create table RVmain2 as select
694         a.*, b.permno
695     from RVmain as a left join home.taq_match as b
696         on a.symbol = b.symbol
697         and a.date = b.table_date;
698 quit;
699
700 * Keep those with matched permno. If no matched permno, bring in most recent date
701 with permno *;
702 data keepers;
703     set RVmain2;
704     if permno ^= '';
705 run;
706
707 data nopermno (drop=permno);
708     set RVmain2;
709     if permno = '';
710 run;
711
712 proc sql;
713     create table nopermno2 as select
714         a.*, abs(intck('weekday',a.date,b.table_date)) as diff, b.permno
715     from nopermno as a left join home.taq_match as b
716         on a.symbol = b.symbol
717     group by a.symbol,a.date having calculated diff=min(calculated diff);
718 quit;
719
720 * Put together and keep only ones with permnos *;
721 data RVmain3;
722     set keepers nopermno2;
723     if permno ^= '';
724 run;
725
726 *****;
727
728 * Now, calculate realized volatility over [0,2] window and bring into main dataset
729 *;
730 proc sql;
731     create table RVmain4 as select distinct
732         a.*, b.symbol, mean(b.RVday) as ave_volat_intra_ea

```

```

733         from home.taqid_merge as a left join RVmain3 as b
734         on a.permno = b.permno
735         and intck('day',a.trading_date,b.date)>=0 and
736 intck('day',a.trading_date_plus2,b.date)<=0
737         group by a.permno, a.fyearqtr
738         order by a.cusip, a.fyearqtr;
739 quit;
740
741 * Same with the estimation window [-41,-11] *;
742 proc sql;
743     create table RVtoUse as select distinct
744         a.*, b.symbol, mean(b.RVday) as ave_volat_intra_est
745     from RVmain4 as a left join RVmain3 as b
746     on a.permno = b.permno
747     and intck('day',a.trading_date_min41,b.date)>=0 and
748 intck('day',a.trading_date_min11,b.date)<=0
749     group by a.permno, a.fyearqtr
750     order by a.cusip, a.fyearqtr;
751 quit;
752
753 * Save down, merge in with Step2_Final dataset in Stata *;
754 data home.RVtoUse; set RVtoUse; run;
755
756
757 *****;
758 **          END OF FILE          **;
759 *****;
760

```