

SECTION IV: SAS CODE TO CLASSIFY TRADES AS PRINCIPAL VS. NOT PRINCIPAL (AGENCY OR EFFECTIVELY AGENT TRADES)

```
/*Part A: Prepare data for classification algorithm*/

/*1A) Create new variables and a unique identifier for each trade-dealer
combination*/;

data bond.aggtradedata121516jff;set bond.aggtradedata121516jff;
entrdr_vol_qt=(100/entrdr_pr)*volume_dol;
quant_sgn=entrdr_vol_qt*txn;
triddc=_N_;
year=year(TRD_EXCTN_DT);
run;

/*2A) Keep only dealer-years in Constant or Top 70% sample*/
data A1;set bond.dealerthresh051917jff;thresh=1;keep year dc thresh;run;
proc sort data=A1;by year dc;run;
proc sort data=bond.aggtradedata121516jff;by year dc;run;
data threshsamp;merge bond.aggtradedata121516jff (in=a) A1 (in=b);by year
dc;if a;run;

data A1;set bond.dealerconstant051917jff;const=1;keep dc const;run;
proc sort data=A1;by dc;run;
proc sort data=threshsamp;by dc;run;
data bothsamp;merge threshsamp (in=a) A1 (in=b);by dc;if a;run;
data bothsamp;set bothsamp;if thresh=1 or const=1;drop thresh const;run;

/*how many cusips?*/
proc sort data=bothsamp nodupkey out=test;by cusip_id;run;
/*xxx cusips*/

/*3A) Create a unique cusip identifier for all cusips*/
proc sort data=bothsamp;by cusip_id dc descending cust trdtime;run;
data bothsamp;set bothsamp;by cusip_id;retain cusid;if _N_=1 then cusid=1;if
first.cusip_id and _N_ ne 1 then cusid=cusid+1;else cusid=cusid;run;
proc sort data=bothsamp nodupkey out=test;by cusid;run;

/*4A) Create sample of potential matched trades. Exclude trades already signed
as A=Agency by FINRA.*/
/*These trades cannot be matched to trades coded as P=Principal in the
effectively agent algorithm.
Round quantity variable so to accurately pick up buys that are exactly offset
by sells and vice versa (to avoid SAS rounding errors)
Keep only necessary variables*/
data bond.algoJFv1;set bothsamp;
if rsc eq 'P';
quant_sgn=round(quant_sgn,.0001);
keep TRD_EXCTN_DT cusip_id volume_dol entrdr_pr trdtime cust txn dc rsc cusid
quant_sgn entrdr_vol_qt triddc;run;
proc sort data=bond.algoJFv1 out=temp nodupkey;by cusip_id;run;

/*Part B: Run all customer trades through the algorithm to identify
"effectively agent" vs. principal trades, defined as: Trades are classified
```

as 'principal' if not reported as 'Agency' by FINRA and not 'reversed' within one minute. Trades are classified as 'reversed' when an exact offsetting quantity (either a customer or interdealer trade) occurs or a combination of 2-3 trades offsets the customer trade within 60 seconds prior or subsequent to the trade. */

/*NOTE: This is a computationally intensive run. For large data sets (like this one), run multiple small data sets at the same time. Below is an example how to cut the data.*/

```
data bond.pair0to4000;set bond.algoJFv1;where cusid le 4000;run;
data bond.pair4000to8000;set bond.algoJFv1;where cusid le 8000 and cusid ge 4001;run;
data bond.pair8000to12000;set bond.algoJFv1;where cusid le 12000 and cusid ge 8001;run;
data bond.pair12000to16000;set bond.algoJFv1;where cusid le 16000 and cusid ge 12001;run;
data bond.pair16000to20000;set bond.algoJFv1;where cusid le 20000 and cusid ge 16001;run;
data bond.pair20000to21000;set bond.algoJFv1;where cusid ge 20001;run;
```

/*1B) Effectively agent algorithm*/

```
proc sort data=bond.algoJFv1;by cusip_id dc TRD_EXCTN_DT trdtime;run;
```

/*Customer trades to be matched*/

```
data cust (rename=(quant_sgn=cqty trdtime=ctim));set bond.algoJFv1;
keep tridcc cusip_id TRD_EXCTN_DT dc quant_sgn trdtime tridcc;where cust=1
and rsc = 'P';;run;
data cust;set cust;custid=_N_;run;
```

/*All trades as potential matches*/

```
data pair1 (rename=(tridcc=ptridcc trdtime=ptrdtime cust=pcust));set
bond.algoJFv1;keep cusip_id TRD_EXCTN_DT dc quant_sgn trdtime cust
tridcc;run;
```

/*2B) Define the total number of observations to send through the algorithm*/

```
data _null_;
  set cust end=eof;
  if eof then call symput('numcust',custid);run;%put &numcust;
proc delete data=bond.algoJFv1OUT;run;
```

/*3B) Begin algorithm*/

```
dm 'log; autoscroll 0';
proc printto log="/home/Bonds/corp/data/Logs/algoJFv1OUT.log";run;
%macro PAIRA;
%do i=1 %to &numcust;
```

```
data cust&i;set cust;where custid=&i;run;
```

```
data P&i;merge cust&i (in=a) pair1 (in=b);by cusip_id dc TRD_EXCTN_DT;if a
and b;
if tridcc ne ptridcc;if abs(ptrdtime-ctim) le 60;run;
```

```
data AP2&i;set P&i;rename quant_sgn=qty2 ptridcc=pid2;keep quant_sgn
ptridcc;run;
```

```

data AP3&i;set P&i;rename quant_sgn=qty3 ptriddc=pid3;keep quant_sgn
ptriddc;run;

data _null_;set P&i end=eof;if eof then call symput('num',_N_);run;%put &num;

proc sql noprint;

select count(*) into :obs_count from P&i;

quit;

%if (&obs_count ne 0) %then %do;

PROC SQL; CREATE TABLE pairv1&i AS SELECT * FROM P&i where cqty+quant_sgn=0
and &num=1;quit;

PROC SQL; CREATE TABLE pairv2&i AS SELECT * FROM P&i,AP2&i where ptriddc ne
pid2 and
(cqty+quant_sgn=0 or cqty+qty2=0 or cqty+quant_sgn+qty2=0) and &num=2;quit;

PROC SQL; CREATE TABLE pairv3&i AS SELECT * FROM P&i,AP2&i,AP3&i where
ptriddc ne pid2 and pid2 ne pid3 and ptriddc ne pid3
and (cqty+quant_sgn=0 or cqty+qty2=0 or cqty+qty3=0 or cqty+quant_sgn+qty2=0
or cqty+quant_sgn+qty2+qty3=0 or cqty+qty2+qty3=0 or cqty+quant_sgn+qty3=0)
and &num ge 3;quit;

data pair2&i;set pairv1&i pairv2&i pairv3&i;if &num=1 then PR=1;if &num=2
then PR=2;if &num ge 3 then PR=3;run;

proc append base=bond.algoJFv1OUT (keep = triddc ptriddc pid2 pid3 pcust PR)
data=pair2&i force;run;
%end;

%end;run;%mend PAIRA;
%PAIRA;
dm 'log; autoscroll 1';
proc printto;run;

/*Part C: Merge with Aggregate data sample then create final Top 70% and
Constant Dealer samples*/

/*1C: Stack data from all runs (Important: see Note at top of Part B)*/;
data allnewpair;set
algoJFv1OUT;
run;

/*2C) Create 'Princ' variable*/
proc sort data=allnewpair (keep=triddc pr) nodupkey;by triddc;run;
proc sort data=bond.aggtradedata121516jf;by triddc;run;
data allnewpair2;merge bond.aggtradedata121516jf (in=a) allnewpair (in=b);by
triddc;if a;run;

data allnewpair2;set allnewpair2;
if pr=1 or pr=2 or pr=3 then rscnewJF='A';
if rscnewJF='A' or rsc='A' then princ=0;else princ=1;run;

```

```
/*3C) Merge with mainthreshsamp010217JF and mainconstsamp010217JF*/
```

```
data one;set allnewpair2;keep trid dc princ;run;  
proc sort data=one;by trid dc;run;  
proc sort data=bond.mainthreshsamp010217JF;by trid dc;run;  
data bond.mainthreshsamp010217JFv2;merge bond.mainthreshsamp010217JF (in=a)  
one (in=b);by trid dc;if a;run;
```

```
data one;set allnewpair2;keep trid dc princ;run;  
proc sort data=one;by trid dc;run;  
proc sort data=bond.mainconstsamp010217JF;by trid dc;run;  
data bond.mainconstsamp010217JFv2;merge bond.mainconstsamp010217JF (in=a) one  
(in=b);by trid dc;if a;run;
```