\*\*\*\* trial of 1:3 matching without replacement, using repeated 1:1 matching with psmatch2 \*\*\*\*  
\*\*\* run your own prediction model, save your propensity score under variable [pscore]  
  
  
\* create copy of propensity score  
sum pscore  
gen pscore\_original=pscore  
  
\* random order  
set seed 1000  
gen x=uniform()  
sort x  
  
  
\*\*\* Round 1  
\*\* nearest neighbour 1:1 matching with caliper 0.20\*SD, adjust for your own data from 'sum pscore' results above   
psmatch2 [your intervention], pscore(pscore) caliper (0.024) noreplacement descending  
  
\*\* remove matched controls by changing propensity score to 91 (future rounds will be 92, 93 etc)  
replace pscore=91 if \_treated==0 & \_weight==1  
  
\*\* keep ID of matched control by generating new n1 and ID variable (new variable without underscore so it doesn't get overwritten)  
gen n1=\_n1  
gen id=\_id  
  
\*\* generate paired ID for later analysis  
gen pair = \_id if pscore==91  
replace pair = \_n1 if \_treated==1  
bysort pair: egen paircount = count(pair)  
replace pair=. if paircount!=2  
drop paircount  
  
  
  
\*\*\* Round 2  
sort x  
\*\* nearest neighbour 1:1 matching with caliper 0.20\*SD   
psmatch2 [your intervention], pscore(pscore) caliper (0.024) noreplacement descending  
  
\*\* remove matched controls by changing propensity score to 92  
replace pscore=92 if \_treated==0 & \_weight==1  
  
\*\* keep ID of matched control by generating new \_n2 variable  
gen \_n2=\_n1  
  
  
gen pair2 = \_id if pscore==92  
replace pair2 = \_n2 if \_treated==1  
gsort pair2 \_treated  
replace pair=pair[\_n+1] if pair==. & pair2!=.  
bysort pair: egen paircount = count(pair)  
drop pair2 paircount  
  
  
\*\*\* Round 3  
sort x  
\*\* nearest neighbour 1:1 matching with caliper 0.20\*logit of SD   
psmatch2 [your intervention], pscore(pscore) caliper (0.024) noreplacement descending  
  
\*\* remove matched controls by changing propensity score to 93  
replace pscore=93 if \_treated==0 & \_weight==1  
  
\*\* keep ID of matched control by generating new n1 variable  
gen \_n3=\_n1  
  
gen pair2 = \_id if pscore==93  
replace pair2 = \_n3 if \_treated==1  
gsort pair2 \_treated  
replace pair=pair[\_n+1] if pair==. & pair2!=.  
bysort pair: egen paircount = count(pair)  
drop pair2  
  
  
\*\*\*\* Tidy up and recreate psmatch 1:3 output  
  
\*create 1:3 match descriptor for all matched  
gen one\_to\_n=(paircount-1)  
replace one\_to\_n=. if one\_to\_n==-1  
drop paircount  
sort \_treated  
by \_treated: tab one\_to\_n  
  
\* reconstruct matches to original ID numbers  
gsort pair pscore  
replace \_n2=id[\_n+2] if pscore[\_n+2]==92 & \_n2!=.  
replace \_n3=id[\_n+3] if pscore[\_n+3]==93 & \_n3!=.  
drop \_n1  
rename n1 \_n1  
  
  
\*\* recreate output from 1:n matching with psmatch2  
replace \_id=id  
replace \_weight=1 if \_treated==1 & \_n1!=.  
replace \_weight=1 if \_treated==0 & one\_to\_n==1  
replace \_weight=0.5 if \_treated==0 & one\_to\_n==2  
replace \_weight=0.333 if \_treated==0 & one\_to\_n==3  
replace \_nn=0 if \_treated==0  
replace \_nn=0 if \_treated==1 & \_n1==.  
replace \_nn=1 if \_treated==1 & \_n1!=. & \_n2==.  
replace \_nn=2 if \_treated==1 & \_n1!=. & \_n2!=. & \_n3==.  
replace \_nn=3 if \_treated==1 & \_n1!=. & \_n2!=. & \_n3!=.  
replace \_support=1 if \_treated==1 & \_weight==1  
replace \_pscore=pscore\_original  
order \_pscore \_treated \_support \_weight \_id \_n1 \_n2 \_n3 \_nn , after (x)  
sort pair  
  
\*\*\*\* check if this worked by using pstest