Tanmay Khole
tanmaykhole208@gmail.com

Macro Code:

SAS DO LOOP

```
%macro cond_int_ph_wt(cond_mvar=, num_cond=, int_mvar=, num_int=);


%do j =1 %to &condcnt. ;
      %let cond = %scan(&cond_mvar., &j , '~');
      %put Condition = &cond. ;
          %do i = 1 %to &intcnt. ;
             %let single = %scan(&int_mvar., &i , '~');
             %put single = &single;
                  proc freq data=ptc_ct_da noprint ;
                     table spnsr*&cond.*&single.*phases  / missing
out=__&cond._&single.(where=(&single. ne " " and &cond. ne " " )
drop=percent) ;
                  run;

                  proc sql noprint ;
                        select count(&cond.) into: cond_chk_cnt
                        from __&cond._&single.
                        ;
                  quit;

                  %if &cond_chk_cnt. > 0 %then %do;
                        data __&cond._&single.;
                        length cond int phases $200.;
                              set __&cond._&single.;
                                    cond=&cond.;
                                    INT=&single. ;
                                    drop &cond. &single. ;
                        run;
                  %end;

                  %else %do;
                        proc datasets lib=work nolist;
                              delete __&cond._&single.;
                        quit;
                        run;
```

SAS CALL EXECUTE

```
%macro cond_int_adv(_cond=, _int=);

      proc freq data=ptc_ct_da noprint;
            table spnsr*&_cond.*&_int.*phases  /
missing out=__&_cond._&_int.(where=(&_int. ne " " and
&_cond. ne " " ) drop=percent) ;
      run;

      proc sql noprint ;
            select count(&_cond.) into: cond_chk_cnt
            from __&_cond._&_int.
            ;
      quit;

      %if &cond_chk_cnt. > 0 %then %do;
            data __&_cond._&_int.;
                  length cond int phases $200.;
                  set __&_cond._&_int.;
                        cond=&_cond.;
                        INT=&_int. ;
                        drop &_cond. &_int. ;
            run;
      %end;

      %else %do;
            proc datasets lib=work nolist;
                  delete __&_cond._&_int.;
            quit;
            run;
      %end;
%mend cond_int_adv;


data _null_;
      set cond_int;
```

Tanmay Khole
tanmaykhole208@gmail.com

```
                    %end;
            %end;
%end;

%mend cond_int_ph_wt;


%cond_int_ph_wt(cond_mvar=&ptc_cond_vars., num_cond=&condcnt.,
int_mvar=&ptc_int_vars., num_int=&intcnt.);
```

```
      mac_str = cats('%nrstr(%', macnm, '(_cond=',
cond, ', _int=', int, '))');

      call execute(mac_str);
run;
```

Pre-processing required for macro to work:

SAS DO Loop

```
proc sql noprint ;
      select distinct name into: ptc_int_vars separated by '~'
      from ptc_ct_cont
      where substr(name,1,3) = "PTC"
      ;
    %let intcnt = &sqlobs;
quit;

%put &ptc_int_vars &intcnt;

proc sql noprint ;
      select distinct name into: ptc_cond_vars separated by '~'
      from ptc_ct_cont
      where substr(name,1,5) = "COND_"
      ;
    %let condcnt = &sqlobs;
quit;

%put &ptc_cond_vars &condcnt;
```

SAS Call Execute

```
proc sql;
      create table cond_int as
      select a.name as cond, b.name as int, "cond_int_adv"
as macnm
      from ptc_ct_cont(where=(substr(name,1,5) = "COND_"))
as a, ptc_ct_cont(where=(substr(name,1,3) = "PTC")) as b
      ;
quit;
```