

*****,

* Project : ZRHM-REXA-07-JP

*

* Program name : t1502010403_ZRHM-REXA-07_V1.sas

*

* Author : W. Yang

*

* Date created : 05/21/2015

*

* Purpose : Create Table 15.2.1.4.3 Summary of Demographics and Other Baseline Characteristics

❖ PP Set

*

* Revision History :

*

* Date Author Ref Revision (Date in YYYYMMDD format)

*

*****,

%let prgname=T1502010403_ZRHM_REXA_07_JP_V1;

options nomprint nosymbolgen nomlogic validvarname=upcase;

options sasautos=("W:\pmp07\macros" sasautos) notes;

%init(delivery=9);

%titlecsv(prgname=&prgname., version=3);

%put &title1;

%put &title2;

%put &APPENDIX;

```
%put &endpoint;
```

```
%put &outname.;
```

```
data adsl;
```

```
    set adam.adsl;
```

```
run;
```

```
%trtrtfg(pgmname=&outname., pgmid=1, new=0, style=, bookmark=%lowcase(&outname.));
```

```
%macro repeat(pop1fl=, sub=, out=);
```

```
data adsl1;
```

```
    set adsl;
```

```
    where &pop1fl.='Y';
```

```
    if   trt01an=4 then do; trt=1; output; end;
```

```
    else if trt01an=5 then do; trt=2; output; end;
```

```
    else if trt01an=3 then do; trt=3; output; end;
```

```
    if &pop1fl.='Y' then do; trt=4; output; end;
```

```
run ;
```

```
proc freq data =adsl1 noprint;
```

```
    table trt/out=treatabt (rename =(count=total)drop=percent);
```

```
run;
```

```
Data _null_;
```

```
    Set treatabt;
```

```
Call symput('n' | strip(put(trt, best.)),strip(put(total, best.)));
```

```
Run;
```

```
%put &n1 &n2 &n3 &n4;
```

```
*** Prepare data for analysis ***;
```

```
proc sort data=sdtm.vs(where=(vstestcd='WSTCIR' and vsstresn>. and visitdy<0)) out=vs; by usubjid  
vsdtc; run;
```

```
data vs(keep=usubjid wstcir);
```

```
set vs;
```

```
by usubjid;
```

```
if last.usubjid;
```

```
wstcir=vsstresn;
```

```
run;
```

```
proc transpose data=adam.adqsnd out=qs1(drop=_) prefix=qs suffix=n;
```

```
where paramn in (23) and avisit='Screening';
```

```
by usubjid;
```

```
id paramn;
```

```
var aval;
```

```
run;
```

```
proc transpose data=adam.adqsnd out=qs2(drop=_) prefix=qs suffix=c;
```

```
where paramn in (23) and avisit='Screening';
```

```
by usubjid;
```

```
id paramn;  
var avalcat1;  
run;
```

```
proc transpose data=adam.adqsnd out=qs3(drop=_) prefix=qs suffix=n;  
  where paramn in (24 25 26 27 28 29);  
  by usubjid;  
  id paramn;  
  var aval;  
run;
```

```
proc transpose data=adam.adqsnd out=qs4(drop=_) prefix=qs suffix=c;  
  where paramn in (24 25 26 27 28 29);  
  by usubjid;  
  id paramn;  
  var avalc;  
run;
```

```
data data1;  
  merge adsl1(in=a) vs qs1 qs2 qs3 qs4;  
  by usubjid;  
  if a;  
run;
```

```
data demo;
```

```
set data1;
```

```
if ucpdgr1="" then do; ucpdgr1n=4; ucpdgr1='Missing'; end;
```

```
height=height/100;
```

```
if prodpref='THS 2.2 menthol' then do; prodpren=1; end;
```

```
else if prodpref='mCC' then do; prodpren=2; end;
```

```
else if prodpref='SA' then do; prodpren=3; end;
```

```
else if prodpref='No preference' then do; prodpren=4; end;
```

```
else if prodpref="" then do; prodpren=5; prodpref='Missing'; end;
```

```
if qs23c='Mild' then qs23cn=1;
```

```
else if qs23c='Moderate' then qs23cn=2;
```

```
else if qs23c='Severe' then qs23cn=3;
```

```
/* if qs24n in (0 5 6) then do; qs24n=5; qs24c='Etc.'; end;*/
```

```
if qs24n=. then do; qs24n=100; qs24c='Missing'; end;
```

```
if qs25n=1 then qs25n=3;
```

```
else if qs25n=2 then qs25n=1;
```

```
else if qs25n=3 then qs25n=2;
```

```
else if qs25n=. then do; qs25n=99; qs25c='Missing'; end;
```

```
if qs26n=. then do; qs26n=99; qs26c='Missing'; end;
```

```
if qs27n=. then do; qs27n=99; qs27c='Missing'; end;
```

```
if qs28n=12 then qs28n=1;
```

```
else if qs28n=13 then qs28n=2;
```

```

else if qs28n>0 then qs28n=qs28n+4;

else if qs28n=. then do; qs28n=3; qs28c='Missing'; end;

if 5<=qs28n<=15 then do; qs30n=4; qs30c='Any amount reported'; end;

if qs28n in (1 2) then do;

    if qs29n=3 then qs29n=1;

    else if qs29n=4 then qs29n=2;

    else if qs29n>0 then qs29n=qs29n+3;

    if 4<=qs29n<=5 then do; qs31n=3; qs31c='Any amount reported'; end;

end;

else do; qs29n=.; qs29c=""; end;

if qs28n in (1 2) then do; qs32n=1; qs32c='Replied "Dont Know" or "Prefer not to Say" to 5a'; end;

run;

```

*** Macro for both Categorical & Continuous variables statistics ***;

```
%macro mdemo(in_dsn=,var_list=,n_max=);
```

*** Macro for Continuous variables statistics ***;

```
%macro mmeans(in_dsn=, class=, var=, order1=);
```

*** Obtain decimal point of data ***;

```
data decimal (keep=&class trt deci);
```

```
    length varChar $200;
```

```
    set &in_dsn;
```

```
    varChar=strip(put(&var, best32.));
```

```
    if index(varChar,'.') ne 0 then deci=length(scan(varChar,2,','));
```

```
    else deci=0;
```

```

        if deci >= 3 then deci=1;

run;

*** Keep only the largest decimal number ***;

proc sort data=decimal;      by &class trt descending deci; run;

proc sort data=decimal nodupkey;  by &class trt;      run;

*** Statistics of Continuous Variables ***;

proc means data=&in_dsn noprint nway;

    class &class trt;

    var &var;

    output out= stat1 n=_n mean=_mean std=_std median=_median min=_min max=_max;

run;

*** Apply decimal points and arrange statistics as required in mock ***;

data stat2 (drop=_:);

    length n meansd median minmax $50 fm1-fm3 $10;

    merge stat1 (in=a)

        decimal;

    by &class trt;

    if a;

    fm1=cats(12,".",strip(put(deci+1,3.)));

    fm2=cats(12,".",strip(put(deci+2,3.)));

    fm3=cats(12,".",strip(put(deci+0,3.)));

```

```

if _deci=0 then _std=ceil(_std*100)/100;

else if _deci=1 then _std=ceil(_std*1000)/1000;

else if _deci=2 then _std=ceil(_std*10000)/10000;


if _mean ne . and _std ne . then meansd=strip(putn(_mean, fm1))||"
("||strip(putn(_std,fm2))||")";

else if _mean ne . and _std eq . then meansd=strip(putn(_mean, fm1))||" (NA)";

if _min ne . then minmax=strip(putn(_min,fm3))||", "||strip(putn(_max,fm3));

if _n ne . then n=strip(putn(_n,best12.));

if _median ne . then median=strip(putn(_median, fm1));

run;


*** Transpose data for final report ***;

proc transpose data=stat2 out=stat3 prefix=col;

by &class;

var n meansd median minmax;

id trt;

run;


*** Prepare Statistic Description & sorting orders as seen in Mock ***;

data final_&order1. (drop=_name_);

length stat1 $150;

set stat3;

if _name_='N' then do; stat1="n"; order2 =6; end;

if _name_='MEANSD' then do; stat1="Mean (SD)"; order2 =7; end;

```



```

if _name_='MEDIAN' then do; stat1="Median"; order2 =8; end;

if _name_='MINMAX' then do; stat1="Min, Max"; order2 =9; end;

order1=&order1;

run;

%mend mmeans;

*** Macro for Categorical variables statistics ***;

%macro mfreq(in_dsn=, n_max=, order1=, order2=, class=);

*** Statistics of Categorical Variables ***;

proc means data=&in_dsn noprint nway;

class &class trt;

var trt;

output out=stat1 n=count;

run;

*** Arrange count & percentage as required in mock ***;

data stat2 (drop=_type_ _freq_ percentx);

set stat1;

by &class trt;

length percentage $25;

%do i=1 %to &n_max;

if trt=&i and count>. then percentx=count/%eval(&&n&i)*100;

if percentx=100 then percentage=put(count,4.)||'(100)';

else if percentx>=0.1 then percentage=put(count,4.)||'('||strip(put(percentx,5.1))||')';

```

```
        else if percentx>. then percentage=put(count,4.)||'(<0.1)';  
    %end;  
run;
```

```
*** Transpose data for final report ***;  
  
proc transpose data=stat2 out=stat3 prefix=col;  
  
    by &class;  
  
    id trt;  
  
    var percentage;  
  
run;
```

```
*** Prepare sorting orders as seen in Mock ***;  
  
data final_&order1 (drop=_name_);  
  
    set stat3;  
  
    order2=&order2;  
  
    order1=&order1;  
  
%do i=1 %to &n_max;  
  
    %if col&i="" %then col&i='0';  
  
%end;  
  
run;  
  
%mend mfreq;
```

```
*** Read in Macro Parameters and call Categorical & Continuous Macros respectively ***;  
  
%local w i x _mtx;  
  
%let w=1;
```

```

%do %while (%scan(&var_list,&w,':') ne );

%local _mtx&w;

%let _mtx&w=%scan(&var_list,&w,':');


%let type=%scan(&&_mtx&w,1, ' ');

*** 'S' is short of 'Summary', stand for Continuous Variable ***;

%if %upcase(&type)=S %then %do;

    %let sum_var=%scan(&&_mtx&w,2, ' ');

    %mmeans(in_dsn=&in_dsn, order1=&w, class=, var=&sum_var);

%end;


/**** 'F' is short of 'Frequency', stand for Categorical Variable ****/

%else %if %upcase(&type)=F %then %do;

    %let fr_var=%scan(&&_mtx&w,2, ' ');

    %let order2=%scan(&&_mtx&w,3, ' ');

    %mfreq(in_dsn=&in_dsn, n_max=&n_max, order1=&w, order2=&order2, class=&order2
&fr_var);

data final_&w.;

    length stat stat1 $150;

    set final_&w.;

    stat="^R/RTF'\li300' " || propcase(strip(&fr_var));

    stat1='n (%)    ';

run;

%end;

```

```

        %let w=%eval(&w.+1);

%end;

%let _mtx=%eval(&w-1);


data final1;

    length col1-col4 $100;

    set final_;;

    keep stat: col: order;;

run;

%mend mdemo;


options symbolgen mprint;

%mdemo(

    in_dsn=demo,

    var_list=F sexc sexn:S age:F ethnic ethnicn:S weightbl:S height:S bmi:F bmigr1 bmigr1n:S wstcir:

        F ucpdgr1 ucpdgr1n:S tarbl:F targr1 targr1n:S nicobl:F nicogr1 nicogr1n:S qs23n:F qs23c qs23cn:

        F prodpref prodpren:F qs24c qs24n:F qs25c qs25n:F qs26c qs26n:F qs27c qs27n:F qs28c qs28n:

        F qs32c qs32n:F qs30c qs30n:F qs31c qs31n:F qs29c qs29n,

    n_max=4);


data shell;

    length stat3 $150;

    order1=1; order2=0; stat3='Sex'; output;

    order1=2; order2=6; stat3='Age (years)'; output;

    order1=3; order2=0; stat3='Ethnicity'; output;

```

```

order1=3; order2=4; stat3="^R/RTF'\li300' " | | 'Non-Japanese'; output;

order1=4; order2=6; stat3='Weight (kg)'; output;

order1=5; order2=6; stat3='Height (m)'; output;

order1=6; order2=6; stat3="BMI (kg/m^{super 2})"; output;

order1=7; order2=1; stat3="^R/RTF'\li300' " | | 'Underweight'; output;

/*  order1=7; order2=1; stat3=""; output; */

order1=8; order2=6; stat3='Waist circumference (cm)'; output;

order1=9; order2=0; stat3='Cigarette consumption'; output;

order1=10; order2=6; stat3='ISO tar yields'; output;

/*  order1=11; order2=1; stat=""; output; */

order1=12; order2=6; stat3='ISO nicotine yields'; output;

/*  order1=13; order2=1; stat=""; output; */

order1=14; order2=6; stat3='FTND score'; output;

/*  order1=15; order2=1; stat=""; output; */

order1=16; order2=0; stat3='Individual product preference'; output;

order1=17; order2=0; stat3="Day 4 SES Questionnaire ^R/RTF'\line' Q1: Total number living in
household (including yourself)?"; output;

order1=18; order2=0; stat3='Q2: Highest level education attained / currently attaining?'; output;

order1=19; order2=0; stat3='Q3: How many income earners in household?'; output;

order1=20; order2=0; stat3='Q4: What is your occupation?'; output;

order1=21; order2=0; stat3='Q5a: What is the approximate MONTHLY household income from all
sources before tax?'; output;

order1=22; order2=1; stat3='Replied "Don T Know" Or "Prefer not to Say" to 5a'; output;

order1=25; order2=0; stat3='Q5b: (For those who answered "Don T know" or "Prefer not to say" to
5a) Would MONTHLY household income be over or under 800,000 yen?'; output;

run;

```

```
data final1;

    set final1;

    if order1=23 then order1=21;

    if order1=24 then order1=25;

    if order1=17 then order2=order2+1;

run;


proc sort data=final1; by order1 order2; run;
```

```
data &out.;

    merge final1 shell;

    by order1 order2;

    if stat3^="" then stat=stat3;


    if order2^=0 then do;

        if col1="" then col1='0'; if col2="" then col2='0';

        if col3="" then col3='0'; if col4="" then col4='0';

    end;
```

```
stat=tranwrd(stat,'To ','to ');

stat=tranwrd(stat,'Ths 2.2 Menthol','THSm2.2');

stat=tranwrd(stat,'Mcc','mCC');

stat=tranwrd(stat,'" Sa"',"' SA'");

stat=tranwrd(stat,'Don T','Don't');

stat=tranwrd(stat,'Or ','or ');
```

```

stat=tranwrd(stat,',000 Yen ','000 Yen - ');

stat=tranwrd(stat,'Yen - or Above','Yen or Above');

stat=tranwrd(stat,'Below 100,000 Yen -','Below 100,000 Yen');

stat=tranwrd(stat,'to which ','To which ');

stat=tranwrd(stat,' Mg',' mg');


if order1=22 then do;

    col1=scan(col1,1,'('); col2=scan(col2,1,'(');

    col3=scan(col3,1,'('); col4=scan(col4,1,'(');

end;


if order1=3 and order2=4 then stat1='n (%)';

if order1=7 and order2=1 then stat1='n (%)';

if order1=21 and order2>4 then stat=tranwrd(stat,"^R/RTF'\li300'", "^R/RTF'\li600'");


if order1 in (7 11 13 15) then order2=order2+10;

if order1 in (7 11 13 15) then order1=order1-1;

if order1=19 and order2=1 then stat=" ^R/RTF'\li300' None - All Are Unemployed";

if order1=25 and order2>=4 then stat=tranwrd(stat,"^R/RTF'\li300'", "^R/RTF'\li600'");

if stat=" ^R/RTF'\li300' Professional/ Technical Worker" then stat=" ^R/RTF'\li300'
Professional/Technical Worker";


pageno=1;

if order1 > 4 then pageno=2;

if order1 > 7 then pageno=3;

if order1 >9 then pageno=4;

```

```

if order1 >13 then pageno=5;

if order1 >16 then pageno=6;

if order1 >17 then pageno=7;

if order1 >19 then pageno=8;

if order1 =20 and order2>12 then pageno=9;

if order1 >20 then pageno=9;

if order1 =21 and order2>8 then pageno=10;

if order1 >21 then pageno=11;

pageno=pageno+%eval((&sub-1)*11);

run;


proc sort; by pageno order1 order2; run;


%let totalpage=44;


%macro reppart(sub=,out=);

%do j = 1 %to 11;

%let i=%eval(&j+(&sub.-1)*11);

proc report data=&out.(where=(pageno=&i.)) headskip headline spacing=4 nowd split='|'
style=[outputwidth=100%] style(header column)=[protectspecialchars=off];

column pageno order1 order2 stat stat1 col1-col4;


define pageno /order order=internal noprint;

define order1 /order order=internal noprint;

define order2 /order order=internal noprint;

```



```

define stat /display "Variable" style(column)=[cellwidth=38% asis=on] style(header)=[just=l];
define stat1 /display "Statistic" style(column)=[cellwidth=10% asis=on] style(header)=[just=l];
define col1 /display "THSm2.2|(N=&n1)" flow style(column)=[cellwidth=12% just=c];
define col2 /display "mCC|(N=&n2)" flow style(column)=[cellwidth=12% just=c];
define col3 /display "SA|(N=&n3)" flow style(column)=[cellwidth=12% just=c];
define col4 /display "Overall PP Set|(N=&n4)" flow style(column)=[cellwidth=12% just=c];

```

```

compute before order1 ;

```

```

line "";

```

```

endcomp;

```

```

compute before _page_/style=[fontweight=bold fontsize=3.75];

```

```

line @1 "&title1 &title2";

```

```

line @1 "Per-Protocol Population - Period &sub.";

```

```

line @1 "^R/RTF\brdrb\brdrs\brdrw30\brsp20\b ' ' ";

```

```

endcomp;

```

```

compute after _page_/style=[fontsize=1.75];

```

```

line @1 "Note: The Overall PP Set refers to all subjects in the Per Protocol Population.";

```

```

line @1 "Note: Height at Screening and body weight and waist circumference at Admission (Day -2)
are used.";

```

```

line @1 "Note: mCC = Menthol conventional cigarettes; SA = Smoking abstinence; THSm2.2 =
Tobacco Heating System 2.2 Menthol.";

```

```

line @1 "Note: BMI = Body Mass Index; FTND = Fagerstr m Test for Nicotine Dependence, SES =
Socio-Economic Status.";

```

```

line @1 "Note: Periods defined as Period 1 ([Day 1   Day 6 confinement]), Period 2 ([Day 6
ambulatory   Day 30 Visit]), Period 3 ([Day 30 Visit   Day 60 Visit]) and Period 4 ([Day 60 Visit   Day
90 Visit]).";

```

```
line @1 "";
```

```
line @1 "&APPENDIX.";
```

```
line @1 "Study ID:ZRHM-REXA-07-JP      Program: &fprgname..sas      Status:  
&repversion./&fdate.      Page: &i of &totalpage.";
```

```
endcomp;
```

```
compute after pageno ;
```

```
line "";
```

```
endcomp;
```

```
run;
```

```
%end;
```

```
%mend;
```

```
%reppart(sub=&sub.,out=&out.);
```

```
%mend;
```

```
%repeat(pop1fl=pprot1fl, sub=1, out=out1);
```

```
%repeat(pop1fl=pprot2fl, sub=2, out=out2);
```

```
%repeat(pop1fl=pprot3fl, sub=3, out=out3);
```

```
%repeat(pop1fl=pprot4fl, sub=4, out=out4);
```

```
ods listing;
```

```
ods rtf close;
```

```
data odata.%sysfunc(scan(&prgname,1,'_'));  
  
  set out1(in=a) out2(in=b) out3(in=c) out4(in=d);  
  
  if a then do; group="PPROT1FL"; end;  
  
  if b then do; group="PPROT2FL"; pageno=pageno+11; end;  
  
  if c then do; group="PPROT3FL"; pageno=pageno+22; end;  
  
  if d then do; group="PPROT4FL"; pageno=pageno+33; end;  
  
run;
```