PharmaSUG 2023 - Paper QT-099

SAS® Macro Design Considerations to Generate Subgroup Table and Forest Plot in Oncology Studies

Yizhuo Zhong, Christine Teng, Merck & Co., Inc., Rahway, NJ, USA

ABSTRACT

Subgroup analyses are essential when there is anticipated heterogeneity within a target population and potential inconsistency of therapeutic response to the study treatment. Subgroup tables and forest plots are commonly required for the efficacy analysis in oncology studies. It is important that the analysis programs can handle additional subgroups which may be added for exploratory analyses or agency requests. This paper will review some common scenarios using categorical variables. These subgroup variables are generally provided through the subject level dataset such as ADSL and efficacy datasets supporting the analysis of overall survival (OS), progression-free survival (PFS), and objective response rate (ORR). The proposed design of a SAS macro would allow flexibility to add additional subgroups, handle different sorting orders within a subgroup, and display text of the output without the need to update the analysis programs. The design also considers the case when the subgroup size is too small for comparison to eliminate unexpected statistical errors when generating the analysis reports in an unblinded environment.

INTRODUCTION

Subgroup tables and forest plots are common deliverables in oncology studies. They are created to analyze and demonstrate the differences in efficacy endpoints across subgroups of subjects. This paper will provide a suggested approach to create subgroup tables and forest plots. Also, the following scenarios when considering the macro design will be covered in this paper:

- Add additional subgroups to the outputs
- Adjust the sorting/display orders within a subgroup
- Adjust the text displayed for a subgroup
- Logic when the subgroup size is too small

The example outputs shown in the paper are created using mock data.

OVERVIEW OF PROGRAMMING FLOW

The proposed macro to create the subgroup tables and forest plots contains two parts. The first part is the data macro to create the analysis result dataset used as the input for the corresponding tables and plots, and the second part is table/plot macros to output subgroup tables and forest plots. **Figure 1** below shows the flow chart of the macro design:

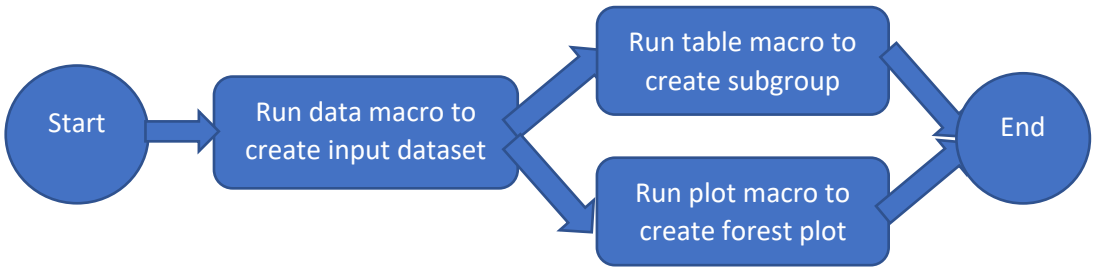


Figure 1 Flow Chart of Macro Design to Create Subgroup Table and Forest Plot

Basically, the data macro includes the %SUMSTAT macro which can calculate the statistical results for both HR and ORR, depending on the analysis or input dataset selected. The codes below show the structure of the %SUMSTAT macro:

```
%if %index(&observation_from, &hr_dataset) %then %do;
    %macro sumstat(inds=, outds=, sortcd=, strata=, sortnm=);
    %* <Codes to calculate hazard ratio (HR)>;
    %mend;
%end;
%else %if %index(&observation_from, &orr_dataset) %then %do;
    %macro sumstat(inds=, outds=, sortcd=, strata=, sortnm=);
    %* <Codes to calculate ORR difference>;
    %mend;
%end;
```

The subgroup variables come from subject level datasets such as ADSL, with the numeric and character variables entered in pairs. The numeric variables control the display order within a subgroup. The following parameters in the data macro will define the labels, variable names (character and numeric) and strata for the subgroups.

```
%create_data(nonstd_section_labels = %str(Age (Years)|Age Group (Years)|
            PD-L1 Status|Gender|Race|ECOG Status|Smoking Status),
            nonstd_section_vars = %str(AGEGR1^AGEGR1N|AGEGR2^AGEGR2N|
            PDL1P10^PDL1P10N|SEX^SEXN|RACEGR1^RACEGR1N|
            ECOGFL1^ECOGFLN1|SMOKER^SMOKERN),
            nonstd_section_strata = %str(stratum|stratum|stratum|
            stratum|stratum|stratum|stratum)
            %* <More macro parameters>;
            );
```

These parameters will be stored in intermediate datasets and parsed into series of macro variables. Then the data macro will loop through each subgroup variable (e.g., SEX) and category (e.g., F/M) based on the order specified in the call, and generate the dataset with statistics calculated from %SUMSTAT macro for each subgroup:

```
%do _k=1 %to &_nonstd_count;
    %let section=%eval(50 + (10 * &_k));

    %* Obtain the numeric and character values of subgroup variables;
    proc freq data=all_merged_2 noprint;
        where &&nonstd_var&_k^='';
        tables &&nonstd_var&_k.*&&nonstd_sort&_k./list missing out=count;
    run;

    %* Create macro variables for the values of subgroups;
    data _null_;
        set count end=last;
        call symput("cat"||strip(put(_n_,best.)), strip(&&nonstd_var&_k));
        call symput("sortcat"||strip(put(_n_,best.)),
                    strip(put(&&nonstd_sort&_k,best.)));
        if last then call symput("ncat",strip(put(_n_,best.)));
    run;

    %* Loop through each subgroup category;
    %do j = 1 %to &ncat;
```

```
    data _&&nonstd_var&_k;
        set all_merged;
        where &&nonstd_var&_k="&&cat&j";
        section=&section;
    run;

    %sumstat(inds=_&&nonstd_var&_k,
            outds=_out_&&nonstd_var&_k.&&sortcat&j,
            sortcd=%eval(&section+&&sortcat&j),
            strata=%str(&&nonstd_strata&_k),
            sortnm="&&cat&j");
    %end;
%end;
```

The second step is to output the subgroup tables or forest plots. There are two macros to create the subgroup tables and forest plots respectively, and the analysis result dataset created by the data macro can be used as the input for both table and plot.

Codes in the table macro to generate second the header of subgroup tables:

```
%if %index(&observation_from, &hr_dataset) %then %do;
    %let header2 =|  |N !Number of Events (%)|N !Number of Events (%)|
                  Hazard Ratio&dagger_.(95% CI)&dagger_.|;
%end;
%if %index(&observation_from, &orr_dataset) %then %do;
    %let header2 =|  |N !Number of Responses (ORR%)|
    N !Number of Responses (ORR%)|Rate Difference&dagger_.(95% CI)&dagger_.|;
%end;
```

Codes in the plot macro to generate the header of forest plots:

```
%if %index(&observation_from, &hr_dataset) %then %do;
    %let plot_header1    = N/#Events|HR|95% CI;
    %let plot_var        = n_trt|est|ci!;
%end;
%else %if %index(&observation_from, &orr_dataset) %then %do;
    %let plot_header1    = N/#Responses|ORR Diff|95% CI;
    %let plot_var        = n_trt|est|ci!);
%end;
```

Figure 2 and **Figure 3** below show the sample outputs of subgroup tables and forest plots.

	Treatment A		Treatment B		Treatment A vs Treatment B Hazard Ratio† (95% CI)†
	N	Number of Events (%)	N	Number of Events (%)	
Overall	326	240 (73.6%)	346	238 (68.8%)	1.13 (0.94,1.36)
Age (Years)					
< 65	103	74 (71.8%)	122	85 (69.7%)	1.06 (0.77,1.46)
>= 65	223	166 (74.4%)	224	153 (68.3%)	1.19 (0.95,1.48)
Age Group (Years)					
< 65	103	74 (71.8%)	122	85 (69.7%)	1.06 (0.77,1.46)
>= 65 and < 75	122	91 (74.6%)	152	100 (65.8%)	1.24 (0.93,1.66)
>= 75 and < 85	95	70 (73.7%)	62	45 (72.6%)	1.04 (0.71,1.51)
>= 85	6	5 (83.3%)	10	8 (80.0%)	1.25 (0.35,4.38)

Figure 2 Sample Output of Subgroup Table

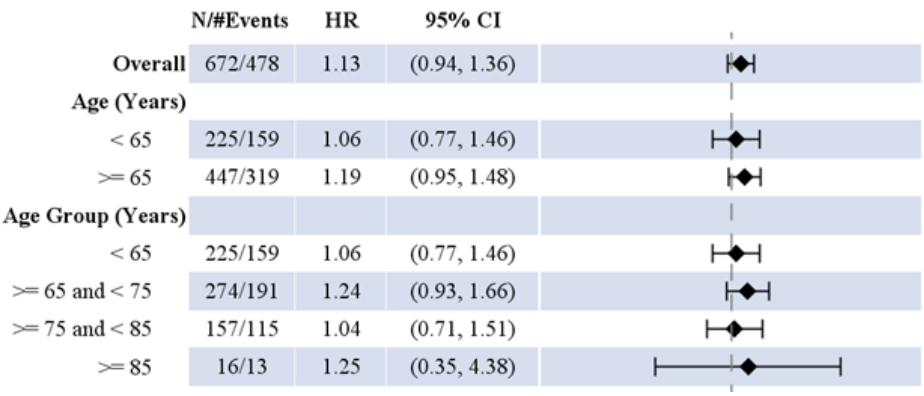


Figure 3 Sample Output of Forest Plots

SAS MACRO DESIGN CONSIDERATIONS

In this section we will propose some design considerations to add flexibilities to the macro logic, which will also enable us to handle different kind of requests for deliverables.

1. ADD ADDITIONAL SUBGROUPS

Sometimes additional subgroups may be requested for analysis purpose. Also, some subgroup variables may come from a different analysis dataset. Suppose that we would like to add "Geographic Region 1" section (EU or non-EU) into our outputs, and the corresponding variables GEOGR1 and GEOGR1N come from ADBASE subject-level dataset. In this case the macro parameters can be adjusted as below:

```
%create_data(nonstd_section_labels = %str(Age (Years)|Age Group (Years)|
                                PD-L1 Status|Gender|Race|ECOG Status|
                                 Geographic Region 1|Smoking Status),
                nonstd_section_vars = %str(AGEGR1^AGEGR1N|AGEGR2^AGEGR2N|
                                PDL1P10^PDL1P10N|SEX^SEXN|RACEGR1^RACEGR1N|
                               ECOGFL1^ECOGFLN1|GEOGR1^GEOGR1N|SMOKER^SMOKERN)
                   population_where = %str(adbase, adsl);
    %* <More macro parameters>;
                );
```

After calling the macro, the updated outputs will show the new subgroup added with the corresponding statistics (see **Figure 4** and **Figure 5**).

Geographic Region 1					
EU	136	61 (44.9%)	150	64 (42.7%)	2.2 (-9.28,13.63)
Non-EU	190	83 (43.7%)	196	94 (48.0%)	-4.3 (-14.13,5.67)
Smoking Status					
Never	120	53 (44.2%)	124	59 (47.6%)	-3.4 (-15.79,9.07)
Former	158	71 (44.9%)	160	78 (48.8%)	-3.8 (-14.68,7.15)

Figure 4 Updated Subgroup Table with New Subgroup Added

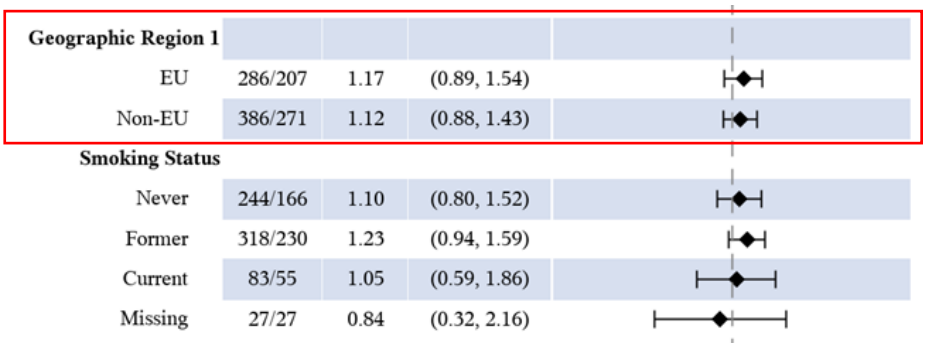


Figure 5 Updated Forest Plot with New Subgroup Added

2. PRE-PROCESS THE INPUT DATASET

Sometimes, the input dataset may need to be pre-processed to obtain the desired results for agency requests, without updating the original ADaM datasets.

In some cases, we may want to adjust the sorting order for the items within a subgroup. For example, in ADSL dataset there are sex values with "M" (Male) decoded as 1 and "F" (Female) decoded as 2. In this way, the outputs will show "M" before "F". Suppose that the order of the sex values needs to be switched. This can be implemented by pre-processing the ADSL dataset as below. Here a numeric variable (SEXN2) with new sets of values is created for reordering purpose, to avoid overwriting the existing SEXN variable:

```
data adsl;
    set adam.adsl;
    if sex='F' then sexn2=1;
    else if sex='M' then sexn2=2;
run;
```

The macro parameter can also be updated to use the new set of numeric values for SEX:

```
%create_data(nonstd_section_vars = %str(AGEGR1^AGEGR1N|AGEGR2^AGEGR2N|
             PDL1P10^PDL1P10N|SEX^SEXN2|RACEGR1^RACEGR1N|
             ECOGFL1^ECOGFLN1|SMOKER^SMOKERN)
             %* <More macro parameters>;
             );
```

After calling the macro, the updated outputs will show that the order within the "Gender" section are switched (see **Figure 6** and **Figure 7**). Note that "Male/Female" values, instead of "M/F", are displayed here. We will discuss about applying format for SEX variable in the following section.

Gender					
Male	241	177 (73.4%)	258	179 (69.4%)	1.12 (0.91,1.38)
Female	85	63 (74.1%)	88	59 (67.0%)	1.22 (0.84,1.78)

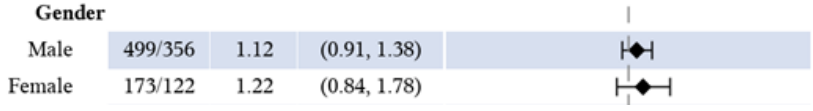


Figure 6 Subgroup Table and Forest Plot Before Switching the Display Order

Gender					
Female	85	63 (74.1%)	88	59 (67.0%)	1.22 (0.84,1.78)
Male	241	177 (73.4%)	258	179 (69.4%)	1.12 (0.91,1.38)

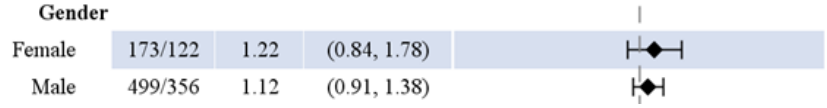


Figure 7 Subgroup Table and Forest Plot After Switching the Display Order

Another example is about handling the “Missing” category within a subgroup. Suppose that we would like to exclude the “Missing” categories in “Race” and “Smoking Status” sections. In this case, the input data can be pre-processed as below by creating new sets of variables:

```
data adsl;
    length racegr2 smoker2 $200;
    set adam.adsl;
    if racegr1='Missing' then do;
        racegr2='';
        racegr2n=.;
    end;
    else do;
        racegr2=racegr1;
        racegr2n=racegr1n;
    end;
    if smoker='Missing' then do;
        smoker2='';
        smoker2n=.;
    end;
    else do;
        smoker2=smoker;
        smoker2n=smokern;
    end;
run;
```

In this way, the original RACEGR1 and SMOKER variables will not be overwritten. The parameter in the data macro can be updated accordingly to use the new set of variables for the corresponding sections:

```
%create_data(nonstd_section_vars = %str(AGEGR1^AGEGR1N|AGEGR2^AGEGR2N|
         PDL1P10^PDL1P10N|SEX^SEXN2|RACEGR2^RACEGR2N|
         ECOGFL1^ECOGFLN1|SMOKER2^SMOKER2N)
         %* <More macro parameters>;
         );
```

As a result, since the macro does not count the missing values when reading the counts and calculating the statistics for each subgroup, the “Missing” category is not displayed for both subgroups. **Figure 8** and **Figure 9** show the forest plot outputs before and after the input dataset is pre-processed.

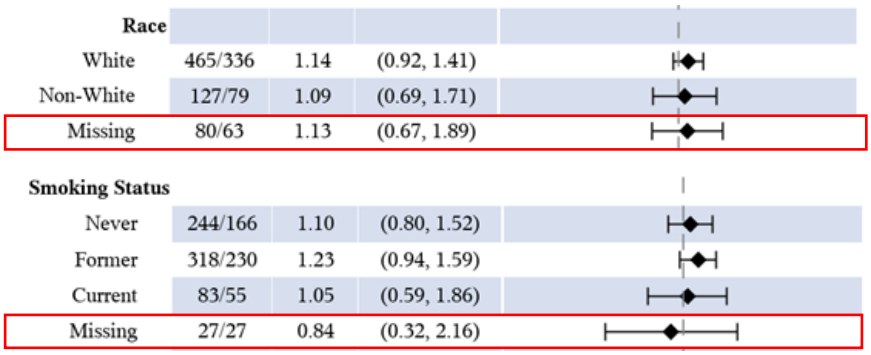


Figure 8 Forest Plot Before Pre-processing of Input dataset

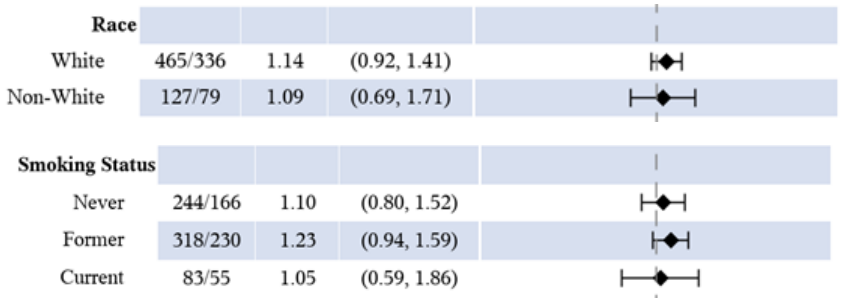


Figure 9 Forest Plot Before Pre-processing of Input dataset

3. ADJUST THE TEXT DISPLAYED FOR A SUBGROUP

Sometimes we may want to adjust the text displayed for a subgroup in our outputs. This can be done by adding pre-specified formats to our categorical variables. Suppose that we would like to change the label for SEX and ECOGFL1 variables in our existing outputs. We first specify the formats for these 2 variables below and load them into SAS libraries:

```
proc format;
    value $sexf
        'F'='Female'
        'M'='Male'
    ;
    value $ecogfl1f
        '[0] Normal Activity or [1] Symptoms, but ambulatory'='0'
        '[2] Ambulatory but unable to work'='1'
        'Missing'='9'
    ;
run;

proc format library=work cntlout=formats;
run;
```

In the data macro, the parameter VAR_FORMAT can specify the variables that need to apply formats. Multiple variables can be separated by comma (,).

```
%create_data(var_format= %str(SEX,ECOGFL1)
            %* <More macro parameters>;
            );
```

Then the macro will read in the FORMATS dataset and merge it with the intermediate datasets, so that the format can be applied to the corresponding variables SEX and ECOGFL1.

The following codes can be included when the macro loops through each category. The logic will apply the formats for the variables specified:

```
%* Apply the formats for subgroups with formats defined;
data all_merged_2;
    set all_merged;
    %if &&nonstd_fmt&_k^=%str() %then %do;
        &&nonstd_var&_k._o=strip(put(&&nonstd_var&_k,$&&nonstd_fmt&_k...));
        drop &&nonstd_var&_k;
        rename &&nonstd_var&_k._o=&&nonstd_var&_k;
    %end;
run;
```

Figure 10 and **Figure 11** below show the labeling changes for "Gender" and "ECOG Status" sections compared with the previous outputs, after applying formats for SEX and ECOGBL1 variables.

Gender					
F	85	63 (74.1%)	88	59 (67.0%)	1.22 (0.84,1.78)
M	241	177 (73.4%)	258	179 (69.4%)	1.12 (0.91,1.38)
Race					
White	225	167 (74.2%)	240	169 (70.4%)	1.14 (0.92,1.41)
Non-White	60	40 (66.7%)	67	39 (58.2%)	1.09 (0.69,1.71)
Missing	41	33 (80.5%)	39	30 (76.9%)	1.13 (0.67,1.89)
ECOG Status					
[0] Normal Activity or [1] Symptoms, but ambulatory	298	215 (72.1%)	325	221 (68.0%)	1.11 (0.92,1.35)
[2] Ambulatory but unable to work	28	25 (89.3%)	21	17 (81.0%)	0.99 (0.49,1.96)

Figure 10 Subgroup Table Before Applying Format

Gender					
Female	85	63 (74.1%)	88	59 (67.0%)	1.22 (0.84,1.78)
Male	241	177 (73.4%)	258	179 (69.4%)	1.12 (0.91,1.38)
Race					
White	225	167 (74.2%)	240	169 (70.4%)	1.14 (0.92,1.41)
Non-White	60	40 (66.7%)	67	39 (58.2%)	1.09 (0.69,1.71)
Missing	41	33 (80.5%)	39	30 (76.9%)	1.13 (0.67,1.89)
ECOG Status					
0	298	215 (72.1%)	325	221 (68.0%)	1.11 (0.92,1.35)
1	28	25 (89.3%)	21	17 (81.0%)	0.99 (0.49,1.96)

Figure 11 Subgroup Table After Applying Format

4. WHEN THE SUBGROUP SIZE IS TOO SMALL

During the analysis, sometimes the size for some subgroups may be too small, which leads to the fact that the results calculated by the statistical procedure may not be meaningful. Therefore, the threshold value for the size of the subgroups to be excluded can be specified. This specific number or percentage for a subgroup to be excluded should be determined per statisticians' guidance, depending on protocol and analysis assumptions.

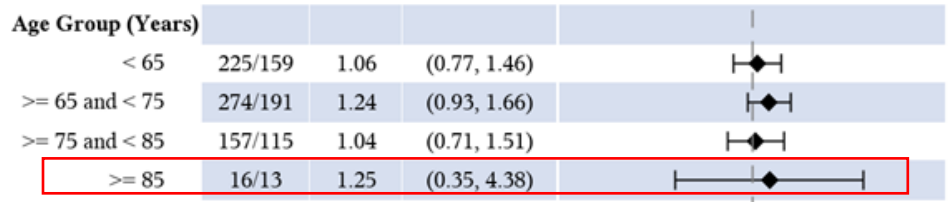
Suppose that the threshold value is set to 30. That means, any category under a subgroup with less than 30 subjects will not be counted in the analysis. The parameter EXCLUDE in the data macro can be used to specify the value as below:

```
%create_data(exclude=%str(<30)
             %* <More macro parameters>;
             );
```

The macro logic to exclude the subgroups with small group size is specified inside the %SUMSTAT macro. The macro will read the counts for each subgroup and parse the EXCLUDE parameter entered above. In this case, when the size for a category under a subgroup is less than 30, the macro will skip the statistics calculation, so the results for this category will not be shown in the outputs:

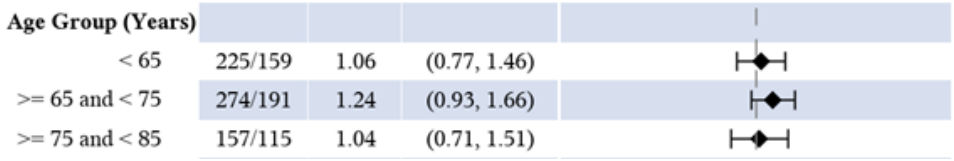
```
proc sql noprint;
    select count(*) into :nobs1 from &inds;
quit;

%let cond = %cmpres(&exclude);
%let exc = %substr(&cond, 2);
%let sign = %substr(&cond, 1, 1);

%if &nobs1. &sign. &exc. %then %goto sumexit;
```

Figure 12 and **Figure 13** below shows the outputs before and after this exclusion threshold is applied. In the original outputs, there are only 16 subjects in the >=85 age group. This triggers the exclusion criteria, and the “>=85” category under “Age Group (Years)” is excluded in the updated output.



Age Group (Years)				
< 65	225/159	1.06	(0.77, 1.46)	
>= 65 and < 75	274/191	1.24	(0.93, 1.66)	
>= 75 and < 85	157/115	1.04	(0.71, 1.51)	
>= 85	16/13	1.25	(0.35, 4.38)	

Figure 12 Forest Plot Before Applying Exclusion Threshold



Age Group (Years)				
< 65	225/159	1.06	(0.77, 1.46)	
>= 65 and < 75	274/191	1.24	(0.93, 1.66)	
>= 75 and < 85	157/115	1.04	(0.71, 1.51)	

Figure 13 Updated Forest Plot with >=85 Age Group Excluded

CONCLUSION

For related efficacy analyses, it is important to communicate with study statistician regarding the requirements (including mockup shells) of the outputs before any programming activities. Depending on study design and requests, the macro may be adjusted accordingly to fit in the analysis settings. The proposed design combined the derivations of HR and ORR in a single macro, and users may consider separate statistical analysis into two macros for easy maintenance and allowing us to incorporate additional analysis. Although the macro design proposed in this paper may not cover all the scenarios for different kind of analysis, utilizing SAS macro programming with careful design to maintain reusability

without constant updates for additional subgroup requirements can streamline the process to create the desired deliverables. Therefore, the proposed macro will help us handle different programming activities based on our analysis needs and will improve our efficiency in the analysis and reporting process.

REFERENCES

Europeans Medicines Agency (EMA). Guideline on the Investigation of Subgroups in Confirmatory Clinical Trials. 2019.

https://www.ema.europa.eu/en/documents/scientific-guideline/guideline-investigation-subgroups-confirmatory-clinical-trials_en.pdf

ACKNOWLEDGEMENTS

The authors would like to give special thanks to the management for their review and inputs on the paper.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:

Yizhuo Zhong
Senior Scientist, Statistical Programming
Merck & Co., Inc.
yizhuo.zhong@merck.com

Christine Teng
Principal Scientist, Statistical Programming
Merck & Co., Inc.
christine_teng@merck.com