PharmaSUG 2019 - Paper DV-131

An Innovative Efficacy Table Programming to Automate Its Figure Generation to Ensure Both High Quality and Efficiency

Xiangchen (Bob) Cui, Letan (Cleo) Lin, and Min Chen, Alkermes Inc., Waltham, MA

ABSTRACT

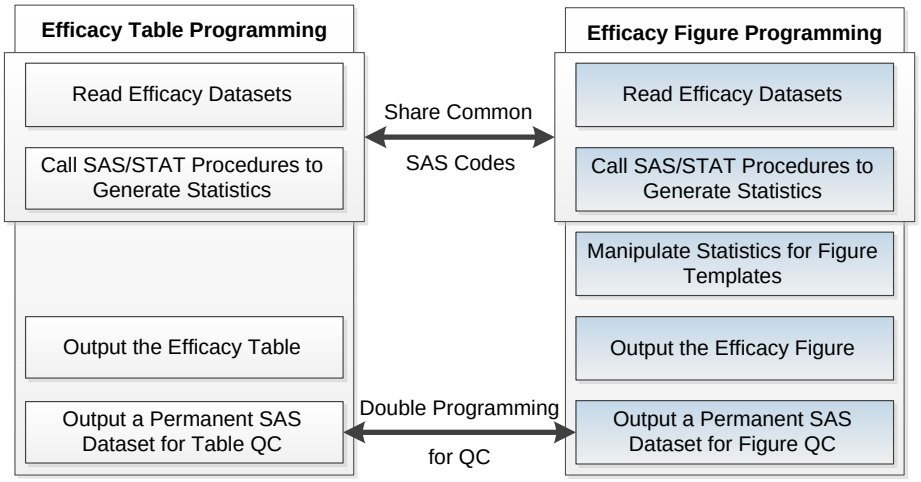
Efficacy table and its figure programming are key part of Statistical Programming to support Clinical Study Report (CSR) [1]. Typically efficacy table programming and its figure programming are two totally independent processes. However they share the common SAS codes to generate statistics for table reporting and figure creation, re the reading of efficacy ADaM datasets to subset the records and select the population, and the calling of SAS Statistical procedures to generate statistics. Hence the consistency between efficacy table programming and its figure programming is very crucial to achieving the quality. Since Statistical Programming for CSR Reporting may undergo a lot of changes until very late in the preparation stage, it requires a lot of resources to maintain the consistency.

This paper presents a new approach to change these two totally independent processes into a "sequential" process by leveraging efficacy table programming from the "common SAS codes" to output extra permanent SAS datasets, which can be directly used in figure programming for the automation of efficacy figure creation. Since the "common SAS codes" in figure programming is removed, the maintenance of the "consistency" can be automatically achieved. Furthermore, the workload for validating figure programming can be dramatically reduced from the double programming to less resource-requiring process, i.e., reviewing production SAS program, its log file, and its output to make sure the SAS program follows SAP and TFL Table Shell. A lot of resources can be saved. There is a growing recognition that the multiple imputation (MI) method can be used to handle missing values in clinical trials. However it requires a lot of computation time. The new method can dramatically reduce SAS program running time for the generation of efficacy tables and their figures when the number of imputations is very big. Hence it helps the programming team's final delivery, especially the key study data readout.

We illustrate the new method by providing examples of forest plots from subgroup analysis of both last visit and all visits to show how this new approach automates the creation of figures efficiently.

INTRODUCTION

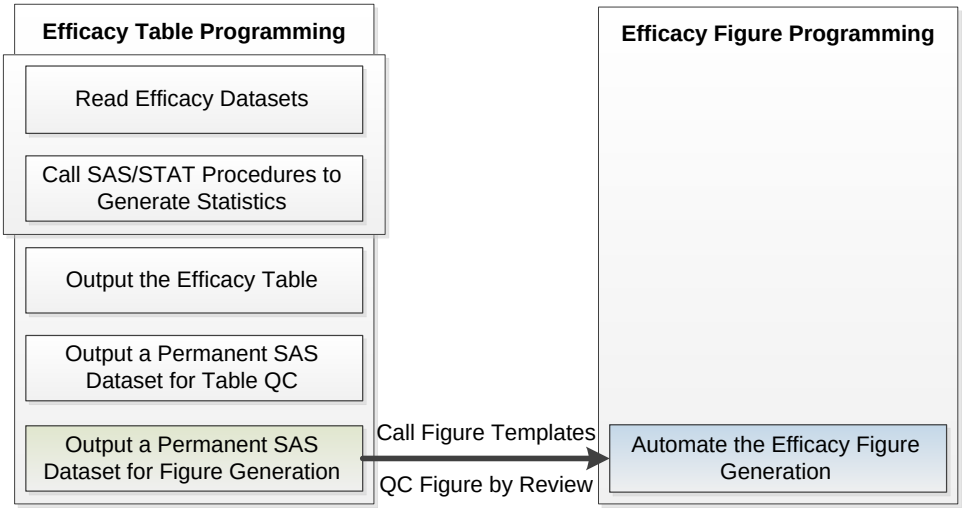
Efficacy table and its figure programming, along with safety table and its figures, are critical part of Statistical Programming for Clinical Study Report (CSR) of a clinical trial [1]. The efficacy table programming and its visualization (figure) programming are normally conducted under two totally independent processes, which can be called ***parallel process***. Below is a flowchart depicting the ***parallel process*** to **independently** generate an efficacy table and its figure.



Display 1. Flowchart of the Parallel Process to Generate an Efficacy Table and its Figure

Four (4) independent SAS programs with the common SAS code for the generation of statistics from the same statistical model for both the table and the figure is needed in the whole programming process of both production and validation. Hence the maintenance of consistency between efficacy table programming and its figure programming with the common SAS codes to generate statistics for table/figure reporting is most important for the quality. To achieve the goal, it requires a lot of efforts (resource and time), for constant changes of reporting are very common in statistical programming until the very later stage of development cycle. Additionally, if the table and its figure programming are developed by two different programmers, both clear communication between them and documentation of programming should be the part of the task.

Hence, an innovative approach is warranted for a high quality and cost-effective efficacy table and its figure programming. This paper presents a new approach to change the parallel process into a "sequential" process by leveraging efficacy table programming from the "common SAS codes" to output extra permanent SAS datasets, which can be directly used in figure programming for the automation of efficacy figure creation. Display 2 shows the flowchart depicting the sequential process.



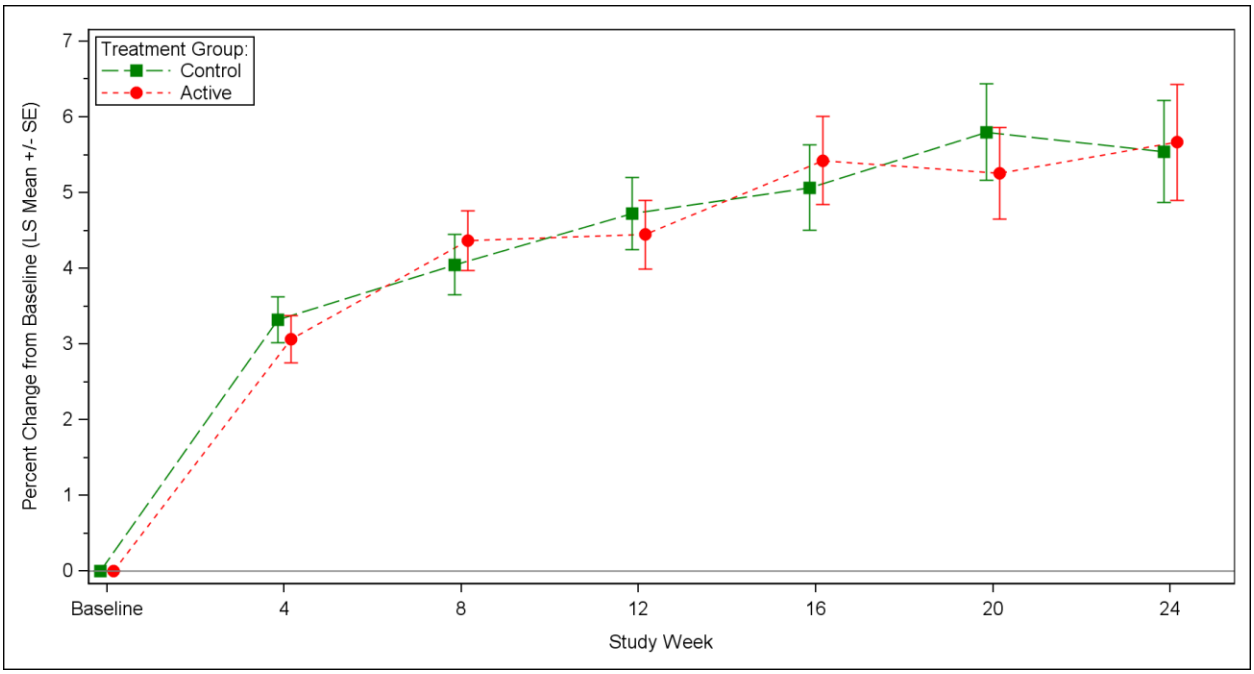
Display 2. Flowchart of the Sequential Process to Generate an Efficacy Table and its Figure

It shows that the new process "skips" the common SAS codes for statistics for both production and validation in figure SAS programming. The maintenance of the "consistency" can be automatically achieved. Furthermore, its validation can be simplified from the double (independent) programming to the review of the production SAS program, its log file, and its output to make sure SAS program follows SAP

and TFL Table Shell, which achieves the cost-effectiveness, in addition to the high quality. The statistics for the figure are normally the part of the efficacy table reporting. Hence the effort to save a permanent SAS dataset for figure generation is very “minimal”.

The multiple imputation (MI) method can be used to handle missing values in clinical trials, and is gaining recognition from both FDA and industry recently. This method requires huge computation time, which raises a big challenge to the statistical programming re the reduction of SAS programming running time for the final delivery, especially to support key study data readout. The “redundancy” of the common SAS code in both table and figure SAS programs from the traditional programming above creates difficulties for the adoption of new methodology re the programming team’s final delivery.

Display 3 shows the figure: “LS Mean of Percent Change from Baseline in Body Weight by Visit (MI) - ANCOVA Approach” as an example of an efficacy figure, which was generated by the new process.



Display 3. An Example of An Efficacy Figure

In this paper, we will use the forest plot programming as an example to illustrate the new methodology by the introduction of forest plot, the standard format of “permanent SAS Dataset for figure generation”, and its figure template SAS program, for forest plot programming is a little complicated.

FOREST PLOT

A forest plot is a graphical display of estimated results from a number of scientific subgroup addressing the same statistics, along with the overall results. And, here are the general elements for a forest plot:

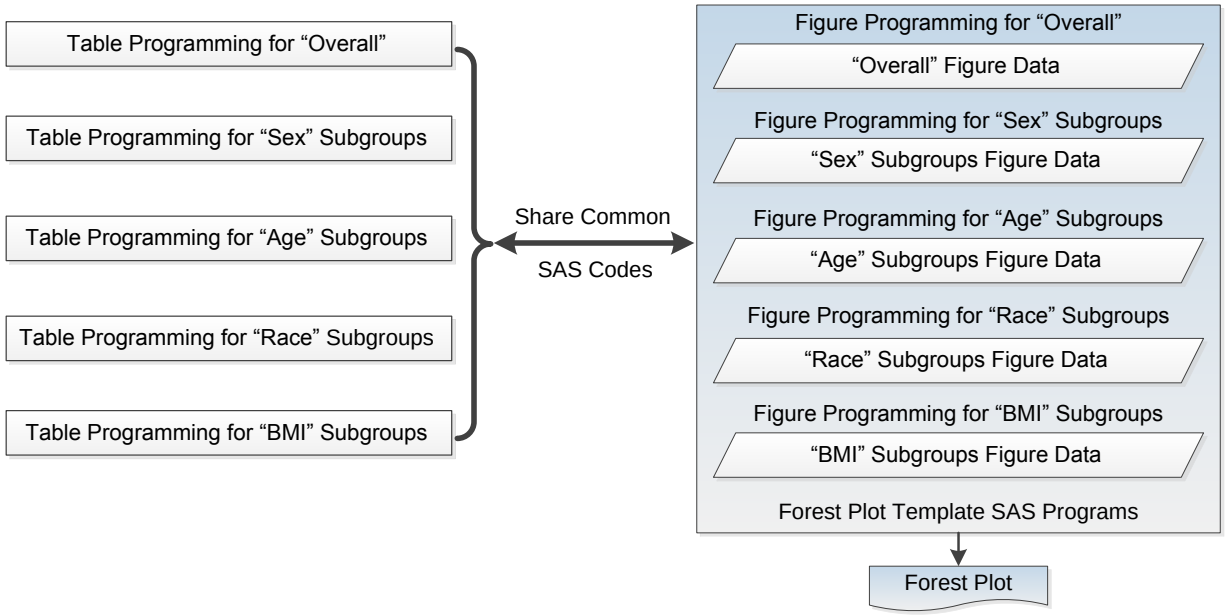
- On the left hand side: subgroup identities and other text
  - Subgroup included in the analysis and incorporated into the forest plot will generally be identified;
  - Other useful text will provide more details for the statistical results.
- On the center: standardized mean difference
  - The chart portion of the forest plot will be on the center and will indicate the mean difference in effect between the treatments in the studies;
  - The vertical line (y-axis) indicates no effect;

- The horizontal distance of a box from the y-axis demonstrates the difference between the treatments in the studies;
- The power is indicated by the weight (size) of the box;
- The thin horizontal lines, sometimes referred to as whiskers, indicate the magnitude of the confidence interval.

- On the right hand side: more precise statistics
  - More precise statistics show up in number form in the text of each line, e.g. confidence interval and p-value in numbers.

Display 11 and 12 show two examples of forest plot of the least squares mean difference along with 95% CIs in percent change in body weight from baseline at Week 24 between active arm and control arm, where the statistics are from the subgroup analysis of subgroup factors: Sex (male, female), Age (<30 years, ≥30 years), Race (Black or African American, Non-Black or African American), and BMI (<27 kg/m2, ≥27 kg/m2), along with the overall model. Hence five (5) independent blocks of SAS codes inside of the forest plot SAS program are needed to generate the statistics for the plot, which are the duplicates of these corresponding efficacy table SAS programs for the common part. It makes the figure SAS program very complicated and very difficult to maintain. Figure validation SAS program has the same challenging task!

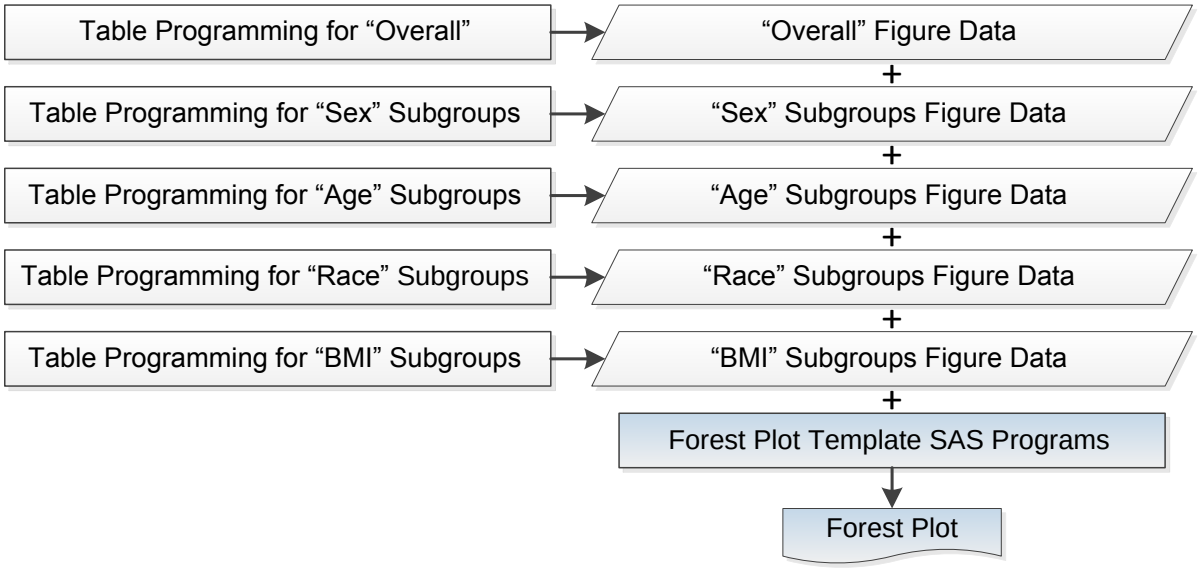
Display 4 shows the parallel process to generate efficacy tables and their figure (forest plot).



Display 4. Flowchart of the Parallel Process to Generate an Efficacy Table and its Figure

Display 5 shows the flowchart of sequential process to generate the forest plot from five efficacy table programs. Note: Each efficacy table programming on the left hand side follows the ***sequential process*** shown in Display 2.

The forest plot SAS program ***simply*** reads these five SAS datasets from the outputs of five efficacy table SAS programs, "piles up" them, and call forest plot template SAS program for the figure generation.



Display 5. Flowchart of Sequential Process to Generate the Forest Plot from Five Efficacy Table Programs

A TEMPLATE FOR A PERMANENT SAS DATASET FOR FIGURE GENERATION

Display 6 shows the variable attributes, along with their roles, for the structure of output dataset for by-subgroupf.

Variable	Type	Len	Label	Function	
SUBGRP	**Char**	**60**	**Subgroup 1**	Subgroup Variables	Used to Automate the Efficacy Figure Generation
SUBGRPN	**Num**	**8**	**Subgroup 1 (N)**		
AVISIT	**Char**	**80**	**Analysis Visit**		
AVISITN	**Num**	**8**	**Analysis Visit (N)**		
SUBTOT1	**Num**	**8**	**Denominator for Trt. Group 1**		
SUBTOT2	**Num**	**8**	**Denominator for Trt. Group 2**		
LCL	**Num**	**8**	**Lower Limit**		
MEAN	**Num**	**8**	**Estimate**		
UCL	**Num**	**8**	**Upper Limit**		
CI	**Char**	**20**	**Confidence Interval**		
PVAL	**Char**	**20**	**P-value**		
NImpute	Num	8	Number of Imputations	Output of SAS/STAT Procedure for Traceability	
Estimate	Num	8	Estimate		
StdErr	Num	8	Std Error		
LCLMean	Num	8	LCLMean		
UCLMean	Num	8	UCLMean		
DF	Num	8	DF		
Min	Num	8	Minimum		
Max	Num	8	Maximum		
Theta0	Num	8	Theta0		
tValue	Num	8	t for H0: Parameter=Theta0		
Probt	Num	8	Pr > |t|		

Display 6. Variable Attributes for a Permanent SAS Dataset for Figure Generation

The variables in bold are directly used in figure template SAS program to automate the forest plot generation. The variables in the second part are directly from SAS/STAT Procedure, and are kept for traceability, further for validation purpose. Note: for the SAS dataset for figure from overall model, **SUBGRP and SUBGRPN are missing.**

ANCOVA APPROACH AND LOGISTIC REGRESSION MODEL APPROACH FOR TABLES AND FOREST PLOTS

There are two different statistical models for the forest plot, and efficacy table, one by ANCOVA approach, another one by logistic regression model approach.

Below shows the SAS codes to get the statistics from ANCOVA model for the subgroup analysis for age group.

```
proc sort data=admiwt2 out=wk24_all;by imputnm avisitn avisit trtpn trtp;
     where fasfl='Y' and avisitn>0;
run;
ods output lsmeans=lsm_all diffs=diff_all;
proc mixed data=wk24_all;
     by imputnm avisitn avisit;
     class trtp(ref='Control') agegr1 racegr1;
     model pchg = base trtp agegr1 trtp*agegr1 racegr1 ;
     lsmeans trtp*agegr1/diff;
run;
*** Combine ANCOVA results by imputation ***;
*** LSM ***;
proc sort data=lsm_all;by agegr1 trtp avisitn avisit;run;
proc mianalyze data=lsm_all;
     by agegr1 trtp avisitn avisit;
     modeleffects estimate;
     stderr stderr;
     ods output ParameterEstimates=pelsm_all;
run;
*** LSMD ***;
proc sort data=diff_all out=diff_all2;by agegr1 avisitn avisit;
     where (trtp ne _trtp) and (agegr1=_agegr1);
run;
proc mianalyze data=diff_all2;
     by agegr1 avisitn avisit;
     modeleffects estimate;
     stderr stderr;
     ods output ParameterEstimates=pdiff_all;
run;
ods pdf close;
ods listing;
run;
```

The highlighted part is used for both the efficacy table and the forest plot generation.

Below shows the SAS codes to manipulate SAS dataset, named **pediff_all,** for the efficacy table.

```
** LSM DIFF for table;
data lsmdiffs;
     length lsmdse lsmdci pval agegr1 $20;
     set pdiff_all;
     lsmdse = strip(put(ESTIMATE,8.2))||" ("||strip(put(STDERR,8.3))||")";
     if n(lclmean, uclmean) > 0 then do;
            lsmdci = "("||strip(put(lclmean,8.2))||
                     ", "||strip(put(uclmean,8.2))||")";
     end;
```

```
      else lsmdci = '-';
      if Probt=. then pval = '-';
      else if Probt>0.999 then pval = ">0.999";
      else if .Z<Probt<0.001 then pval = "<0.001";
      else pval = strip(put(Probt, 5.3));
run;
```

Below shows the SAS codes for outputting of a SAS dataset for forest plot generation.

```
data lsmd_wt_pchg_ancova_mi_fas_age;
     length subgrp $60 ci $20.;
     set lsmdiffs;*** leverage the programming for efficacy figure;
     subgrp=strip(agegr1);
     subgrpn=agegr1n;
     ci=strip(put(ESTIMATE,8.2))||' '||strip(lsmdci);
     mean=estimate;
     lcl=lclmean;
     ucl=uclmean;
     subtot1=count1;
     subtot2=count2;
     label subgrp='Subgroup 1'
           subgrpn='Subgroup 1 (N)'
           subtot1='Denominator for Trt. Group 1'
           subtot2='Denominator for Trt. Group 2'
           lcl='Lower Limit'
           ucl='Upper Limit'
           Mean='Estimate'
           ci='Confidence Interval'
           pval='P-value';
run;
```

Display 7 shows an example of the standardized SAS dataset from the subgroup analysis ANCOVA by age group. Similarly the other four (4) SAS datasets from other subgroup factors: race group, gender, baseline BMI group, and overall model follow the same of the standardized SAS dataset structure as one shown below.

SUBGRP	SUBGRPN	AVISIT	AVISITN	SUBTOT1	SUBTOT2
<30 Years	1	V3-Week 1	8	48	41
<30 Years	1	V4-Week 2	15	48	41
<30 Years	1	V6-Week 4	29	48	41
<30 Years	1	V8-Week 6	43	48	41
<30 Years	1	V9-Week 8	57	48	41
<30 Years	1	V11-Week 12	85	48	41
<30 Years	1	V13-Week 16	113	48	41
<30 Years	1	V15-Week 20	141	48	41
<30 Years	1	V17-Week 24	169	48	41
>=30 Years	2	V3-Week 1	8	200	203
>=30 Years	2	V4-Week 2	15	200	203
>=30 Years	2	V6-Week 4	29	200	203
>=30 Years	2	V8-Week 6	43	200	203
>=30 Years	2	V9-Week 8	57	200	203
>=30 Years	2	V11-Week 12	85	200	203
>=30 Years	2	V13-Week 16	113	200	203
>=30 Years	2	V15-Week 20	141	200	203
>=30 Years	2	V17-Week 24	169	200	203

LCL	MEAN	UCL	CI	PVAL
-0.86330126	0.0276550782	0.9186114164	0.03 (-0.86, 0.92)	0.951
-1.564475376	-0.390068815	0.7843377466	-0.39 (-1.56, 0.78)	0.515
-1.965139182	-0.330285359	1.3045684644	-0.33 (-1.97, 1.30)	0.692
-2.889929401	-0.909219376	1.0714906488	-0.91 (-2.89, 1.07)	0.368
-2.272065501	-0.053753082	2.164559338	-0.05 (-2.27, 2.16)	0.962
-1.970861446	0.5399858659	3.0508331773	0.54 (-1.97, 3.05)	0.673
-2.705053231	0.2597655075	3.2245842465	0.26 (-2.71, 3.22)	0.863
-3.768477106	-0.459970398	2.8485363088	-0.46 (-3.77, 2.85)	0.785
-4.440853438	-0.593348117	3.2541572043	-0.59 (-4.44, 3.25)	0.761
-0.562122389	-0.146539341	0.2690437074	-0.15 (-0.56, 0.27)	0.489
-0.399594562	0.1494114332	0.6984174283	0.15 (-0.40, 0.70)	0.594
-0.903467748	-0.138491659	0.6264844307	-0.14 (-0.90, 0.63)	0.723
-1.272462049	-0.387230829	0.4980003904	-0.39 (-1.27, 0.50)	0.391
-1.266563143	-0.280517131	0.7055288819	-0.28 (-1.27, 0.71)	0.577
-1.631221754	-0.499339857	0.6325420404	-0.50 (-1.63, 0.63)	0.387
-1.753485101	-0.418913327	0.9156584473	-0.42 (-1.75, 0.92)	0.538
-2.197363257	-0.707240585	0.782882086	-0.71 (-2.20, 0.78)	0.352
-1.807886741	-0.158544889	1.4907969623	-0.16 (-1.81, 1.49)	0.850

Display 7. An Example of the Standardized SAS Dataset from the Subgroup Analysis ANCOVA by Age Group

The logistic regression model approach is demonstrated by the following SAS codes.

```
**** logistic regression ****;
ods output diffs=logistic;
proc sort data=admiwt2;by avisitn imputnm;run;
proc genmod data=admiwt2 descending;
     by avisitn imputnm;
     class trtp (ref='Control') agegr1 racegr1/param=glm;
     model myaval=trtp agegr1 trtp*agegr1 racegr1 base/link=logit dist=bin;
     lsmeans trtp*agegr1/diff cl exp;
     where avisitn>15;
run;
proc sort data=logistic out=logistic2;by agegr1 avisitn;
     where (trtp ne _trtp) and (agegr1=_agegr1);
run;
*** combine results ***;
proc mianalyze data=logistic2;
     by agegr1 avisitn;
     modeleffects estimate;
     stderr stderr;
     ods output ParameterEstimates=p_diff;
run;
ods pdf close;
ods listing;
*** back-transformation of Odds ratio for efficacy table ***;
data p_diff2;
     length odds lsmdci teststat pval agegr1 $20 avisit $80.;
     set p_diff;
     if avisitn<=15 then delete;
     odds=strip(put(exp(estimate),8.2));
     if n(lclmean, uclmean) > 0 then do;
```

```
            lsmdci="("||strip(put(exp(lclmean),8.2))||
                   ","||strip(put(exp(UCLmean),8.2))||")";
      end;
      else lsmdci = '-';
      if Probt=. then pval = '-';
      else if Probt>0.999 then pval = ">0.999";
      else if .Z<Probt<0.001 then pval = "<0.001";
      else pval = strip(put(Probt, 5.3));
run;
```

SAS codes are for the outputting of a SAS dataset for forest plot generation from ***logistic regression model***.

```
data Odds_wt_pchg_10pct_mi_fas_age;
      length subgrp $60 ci $20.;
      set p_diff2;
      if agegr1='<30 Years' then agegr1n=1;
      else if agegr1='>=30 Years' then agegr1n=2;
      subgrp=strip(agegr1);
      subgrpn=agegr1n;
      ci=strip(odds)||' '||strip(lsmdci);
      mean=exp(estimate);
      lcl=exp(lclmean);
      ucl=exp(UCLmean);
      label subgrp='Subgroup 1'
          subgrpn='Subgroup 1 (N)'
          subtot1='Denominator for Trt. Group 1'
          subtot2='Denominator for Trt. Group 2'
          lcl='Lower Limit'
          ucl='Upper Limit'
          Mean='Estimate'
          ci='Confidence Interval'
          pval='P-value';
run;
```

Display 8 shows an example of the standardized SAS dataset from the subgroup analysis of logistic regression model by age group. It also follows the same standardized SAS dataset structure shown in **Display 6.** The standardization of this SAS dataset for the figure facilitates the utilization of forest plot template SAS programs to maximize the automation of the figure generation.

SUBGRP	SUBGRPN	AVISIT	AVISITN	SUBTOT1	SUBTOT2
<30 Years	1	V6-Week 4	29	48	41
<30 Years	1	V8-Week 6	43	48	41
<30 Years	1	V9-Week 8	57	48	41
<30 Years	1	V11-Week 12	85	48	41
<30 Years	1	V13-Week 16	113	48	41
<30 Years	1	V15-Week 20	141	48	41
<30 Years	1	V17-Week 24	169	48	41
>=30 Years	2	V6-Week 4	29	200	203
>=30 Years	2	V8-Week 6	43	200	203
>=30 Years	2	V9-Week 8	57	200	203
>=30 Years	2	V11-Week 12	85	200	203
>=30 Years	2	V13-Week 16	113	200	203
>=30 Years	2	V15-Week 20	141	200	203
>=30 Years	2	V17-Week 24	169	200	203

LCL	MEAN	UCL	CI	PVAL
0.2262749675	1.1130022631	5.4746402179	1.11 (0.23, 5.47)	0.895
0.2266210262	1.0111773961	4.5118484514	1.01 (0.23, 4.51)	0.988
0.5141755541	1.8118202418	6.3843809029	1.81 (0.51, 6.38)	0.354
0.2696164697	0.8817701612	2.8837949634	0.88 (0.27, 2.88)	0.834
0.2781457898	0.8762217962	2.7602957315	0.88 (0.28, 2.76)	0.821
0.2646197318	0.7752540308	2.2712547101	0.78 (0.26, 2.27)	0.642
0.2583417068	0.6993379752	1.8931267801	0.70 (0.26, 1.89)	0.481
0.2077730136	0.593289163	1.6941181376	0.59 (0.21, 1.69)	0.329
0.3650913691	0.7790950978	1.662567847	0.78 (0.37, 1.66)	0.518
0.3211973868	0.6397419708	1.2742002455	0.64 (0.32, 1.27)	0.204
0.5684368468	1.0492437183	1.9367364848	1.05 (0.57, 1.94)	0.878
0.5365906282	0.9215856225	1.5828082247	0.92 (0.54, 1.58)	0.767
0.5085949834	0.8708050881	1.4909732228	0.87 (0.51, 1.49)	0.613
0.5589103206	0.9394268569	1.579006125	0.94 (0.56, 1.58)	0.813

Display 8. An Example of the Standardized SAS Dataset from the Subgroup Analysis of Logistic Regression Model by Age Group

CALLING EACH SAS PERMANENT DATASET FOR FOREST PLOT

Five SAS datasets from the subgroup analysis by sex, age group, race group, and baseline BMI group, along with overall analysis, are generated by each efficacy table program described above. Among these five SAS datasets, the records from V17-Week 24 are selected and combined to a single SAS dataset as the input of forest plot template SAS program.

Below shows the SAS codes for the data programming.

```
data allgroup;
    length GROUPN 8 GROUP TEXTVS $50;
    set lsmd_wt_pchg_ancova_mi_v_fas(in=a)
          lsmd_wt_pchg_ancova_mi_fas_sex(in=b)
          lsmd_wt_pchg_ancova_mi_fas_age(in=c)
          lsmd_wt_pchg_ancova_mi_fas_race(in=d)
          lsmd_wt_pchg_ancova_mi_fas_bmi(in=e);
    where avisit='V17-Week 24';
    if a then do; groupn=1; group='Overall'; subgrpn=1; subgrp=''; end;
    else if b then do; groupn=2; group='Sex'; end;
    else if c then do; groupn=3; group='Age'; end;
    else if d then do; groupn=4; group='Race'; end;
    else if e then do; groupn=5; group='BMI'; end;
    textvs='Active (n='||put(subtot2, 3.)||') vs Control (n='||
           put(subtot1, 3.)||')';
    keep groupn group subgrpn subgrp textvs mean lcl ucl ci subtot1 subtot2
         pval;
run;
```

Display 9 shows the “all-group” SAS dataset, which is used as the input for the figure template. The variables with light blue background color are newly assigned based on the figure requirement.

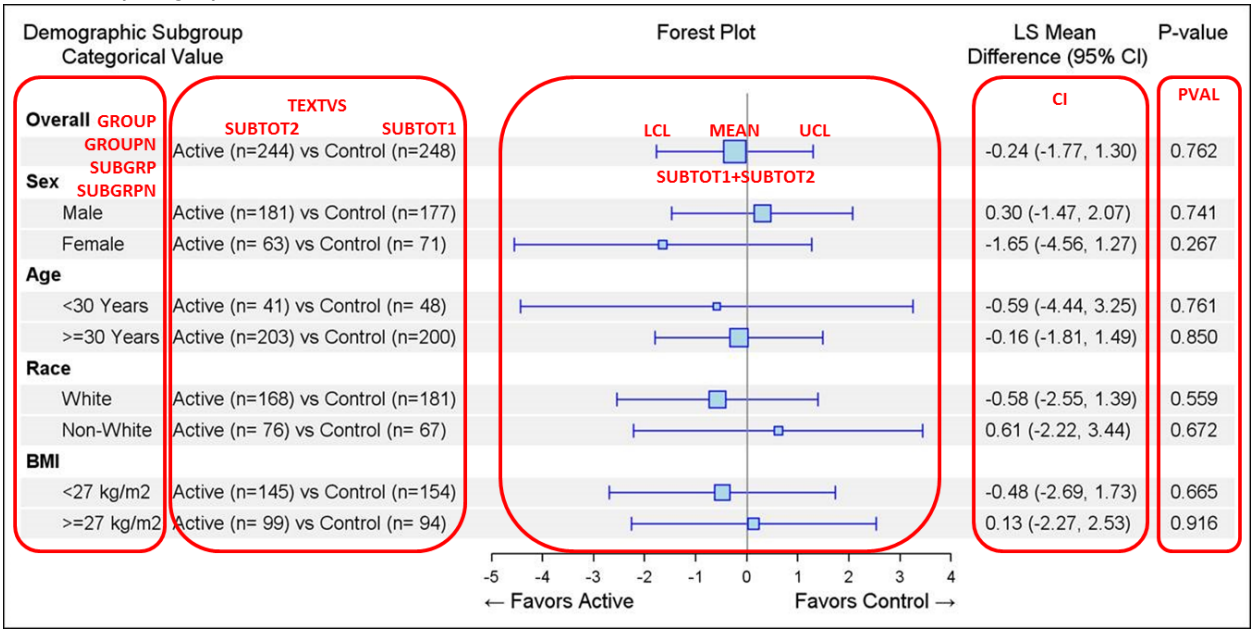
GROUP	GROUPN	SUBGRP	SUBGRPN	TEXTVS	SUBTOT1	SUBTOT2
Overall	1		1	Active (n=244) vs Control (n=248)	248	244
Sex	2	Male	1	Active (n=181) vs Control (n=177)	177	181
Sex	2	Female	2	Active (n= 63) vs Control (n= 71)	71	63

GROUP	GROUPN	SUBGRP	SUBGRPN	TEXTVS	SUBTOT1	SUBTOT2
Age	3	< 30 Years	1	Active (n= 41) vs Control (n= 48)	48	41
Age	3	> =30 Years	2	Active (n=203) vs Control (n=200)	200	203
Race	4	White	1	Active (n=168) vs Control (n=181)	181	168
Race	4	Non-White	2	Active (n= 76) vs Control (n= 67)	67	76
BMI	5	< 27 kg/m2	1	Active (n=145) vs Control (n=154)	154	145
BMI	5	> =27 kg/m2	2	Active (n= 99) vs Control (n= 94)	94	99

LCL	MEAN	UCL	CI	PVAL
-1.77164526	-0.23666796	1.29830935	-0.24 (-1.77, 1.30)	0.762
-1.47293672	0.297186262	2.06730924	0.30 (-1.47, 2.07)	0.741
-4.56307594	-1.64702521	1.26902551	-1.65 (-4.56, 1.27)	0.267
-4.44085344	-0.59334812	3.2541572	-0.59 (-4.44, 3.25)	0.761
-1.80788674	-0.15854489	1.49079696	-0.16 (-1.81, 1.49)	0.85
-2.55154321	-0.5828332	1.38587681	-0.58 (-2.55, 1.39)	0.559
-2.21841314	0.609890779	3.4381947	0.61 (-2.22, 3.44)	0.672
-2.69442029	-0.48261537	1.72918955	-0.48 (-2.69, 1.73)	0.665
-2.27084633	0.129439788	2.52972591	0.13 (-2.27, 2.53)	0.916

Display 9. An Example of a SAS Dataset from ANCOVA Approach for the Forest Plot

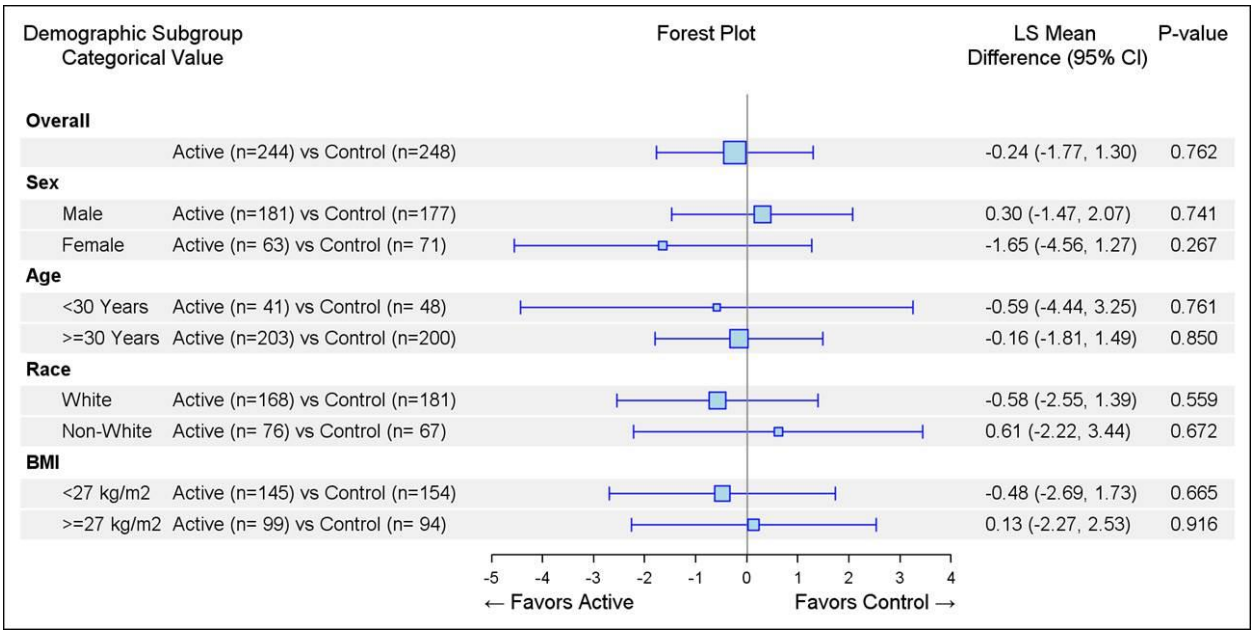
The following is a brief annotation about how to use the standardized SAS data for the forest plotting. It shows one-to-one relationship between each variable of the standardized SAS data and each "field" of the forest plot graph.



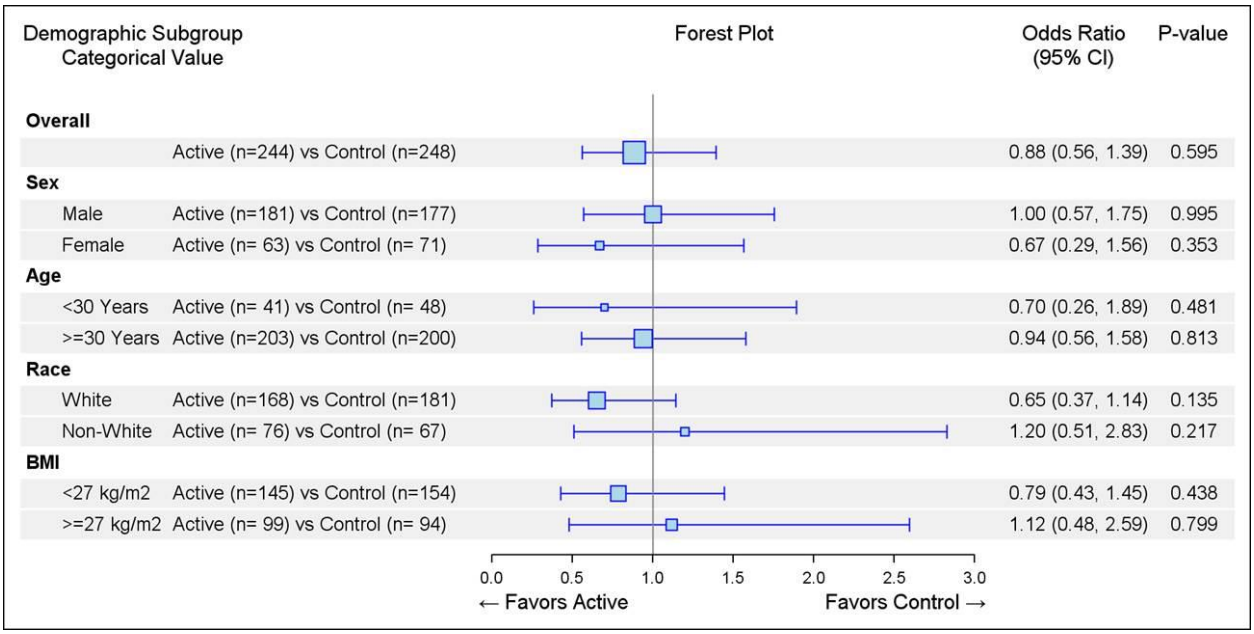
Display 10. Annotation of Forest Plot with the Variables from the Standardized SAS Dataset

The template SAS codes, shown in Appendix 1, could be used for all forest plot programming with minimum changes, which was dedicatedly developed for forest plots. The highlighted SAS codes in Appendix 1 show the place for the minimum changes.

Display 11 and 12 show these two forest plots from two different statistical models by using the same template SAS program.



Display 11. An Example of the Forest Plot from ANCOVA Approach



Display 12. An Example of the Forest Plot from Logistic Regression Model

Appendix 2 shows an example of the forest plot by visits. It was generated by using the same standardized SAS datasets and forest plot template SAS program. It calls the number of the visits, nine (9) in this example, to generate nice (9) forest plots, and output them into one single file. For the presentation purpose, only three visits from visit 3,9, and 17,were chosen among nine plots.

RUN-TIME COMPARISON BETWEEN PARALLEL PROCESS AND SEQUENTIAL PROCESS

Display 13 shows both the Real time and CPU time of five SAS programs for "overall" and subgroup by-visit efficacy tables from Logistic Regression model, which were for both production and validation of SAS programs. 500 imputations (NIMPUTE=500) were used for MI dataset, and efficacy tables. The total time was about 98 hours from production real time (56 hours and 44 minutes) and validation real time (41 hours and 11 minutes), and 87 hours and 40 minutes from production CPU time (52 hours and 58 minutes) and validation CPU (34 hours and 42 minutes).

If the traditional method, i.e., the **parallel process**, was used to generate forest plot shown by Display 12, it took at least about another 98 hours real run-time, or 87 hours and 40 minute CPU time to run forest plot SAS program with five blocks for "overall" and subgroups to generate efficacy data for the figure, plus the time for running the figure template SAS program for figure generation with 7 second real time and 2 second CPU time from this example, which can be "ignored" in terms of rerunning SAS programs. Please refer to **Display 4.** It doubles the run-time from rerunning five efficacy table SAS programs and one forest plot SAS program.

It is easily understood that the new methodology, i.e., **Sequential Process,** only needs the time from rerunning both production and validation of SAS programs for five efficacy tables, for the calling the figure template SAS program for generation of the forest plot was "close to zero" re the run-time! Hence it significantly reduced 98 hours (more than 4 days) for real run-time and 87 hours and 40 minutes for CPU.

In general, there should be about 50% efficiency increase from the new method regarding rerunning SAS programs for final delivery.

This further demonstrates new method facilitates MI analysis by substantially reducing SAS programs run-time, in addition to enhancing the technical accuracy and quality of deliverables.

Program	Real Time		CPU Time	
	PD Running Time	QC Running Time	PD Running Time	QC Running Time
t-mi-v-fas	13 hrs. 13 mins.	10 hrs. 50 mins.	12 hrs. 22 mins.	6 hrs. 40 mins.
t-mi-fas-sex	12 hrs. 54 mins.	8 hrs. 37 mins.	12 hrs. 04 mins.	7 hrs. 50 mins.
t-mi-fas-age	9 hrs. 53 mins.	8 hrs. 8 mins.	9 hrs. 10 mins.	7 hrs. 48 mins.
t-mi-fas-race	12 hrs. 26 mins.	6 hrs. 30 mins.	11 hrs. 32 mins.	6 hrs. 11 mins.
t-mi-fas-bmi	8 hrs. 18 mins.	7 hrs. 6 mins.	7 hrs. 50 mins.	6 hrs. 13 mins.
Total	56 hrs. 44 mins.	41 hrs. 11 mins.	52 hrs. 58 mins.	34 hrs. 42 mins.

Display 13. An Example of Both the Real Run-time and CPU Time Used from the overall and Subgroup Analysis of Logistic Regression Model by Sex Group, Age Group, Race Group, and Baseline BMI Group.

CONCLUSION

This paper presents a new approach to leverage the efficacy table programming to automate its figure generation by outputting extra permanent SAS datasets from the statistical model inside the table SAS program, which is for both the table and the figure. This less-effort step guarantees the consistency between efficacy table programming and its figure programming to achieve the quality. It also simplifies the validation of figure programming by reducing the double programming to the less resource-requiring process, i.e., reviewing production SAS program, its log file, and its output to make sure SAS program follows SAP and TFL Table Shell. It facilitates the development of the multiple imputation (MI) method for the efficacy analysis of a clinical trial by dramatically reducing SAS programming running time for the final delivery. This new technique ensures the technical accuracy and quality, in addition to cost-effectiveness and efficiency. We hope that it can make your life a little easier when you are working on efficacy tables and their figures.

REFERENCES

[1] Guideline for Industry, “Structure and Content of Clinical Study Reports”. Available at https://www.fda.gov/downloads/drugs/guidancecomplianceregulatoryinformation/guidances/ucm073113.pdf.

ACKNOWLEDGMENTS

Appreciation goes to Jerald Schindler for his valuable review and comments.

CONTACT INFORMATION

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

Name: Xiangchen (Bob) Cui, Ph.D.
Enterprise: Alkermes, Inc.
Address: 852 Winter Street
City, State ZIP: Waltham, MA 02451
Work Phone: 781-609-6038
Fax: 781-609-5855
E-mail:xiangchen.cui@alkermes.com

Name: Letan (Cleo) Lin, M.S.
Enterprise: Alkermes, Inc.
Address: 852 Winter Street
City, State ZIP: Waltham, MA 02451
Work Phone: 781-609-6380
Fax: 781-609-5855
E-mail: letan.lin@alkermes.com

Name: Min Chen, Ph.D.
Enterprise: Alkermes, Inc.
Address: 852 Winter Street
City, State ZIP: Waltham, MA 02451
Work Phone: 781-609-6047
Fax: 781-609-5855
E-mail: min.chen@alkermes.com

APPENDIX 1 FULL SAS CODES FOR FOREST PLOT TEMPLATE

```
proc sort data=allgroup;
    by groupn group subgrpn subgrp;
run;

data allgroup;
    set allgroup;
    _marker=_n_;
run;

data shell(keep=groupn group subgrpn subgrp);
    set allgroup;
    subgrpn=0;
    subgrp=group;
run;

proc sort data=shell nodupkey;
    by groupn group subgrpn subgrp;
run;

data figure;
    merge shell(in=a) allgroup(in=b);
    by groupn group subgrpn subgrp;
    _obs=_n_;
    if a then _id=1;
        else if b then _id=2;
    _indent=(_id-1)*2;
    if _marker>. then _ref=_obs;
    length _marfclr _maroclr _marsz _marsbl $20;
    if _marker>. then do;
        _marfclr='lightblue'; _maroclr='blue'; _marsz=strip(put(3+(subtot1+subtot2)/50, 8.1))||'pt';
_marsbl='squarefilled';
    end;
    _mean=mean;
    if _marker>. and mean=. then do;
        _mean=0; _marsz='0pt';
    end;
    _blank=' ';
run;
```

```
*** get macros for figure output ***;
*** figure labels;
%let hl_lbl1=Demographic Subgroup;
%let hl_lbl2=%str(       Categorical Value);
%let hc_lbl1=Forest Plot;
%let hc_lbl2=;
%let hr_lbl1=LS Mean;
%let hr_lbl2=Difference (95% CI);
%let hr2_lbl1=P-value;
%let hr2_lbl2=;

*** marker information;
proc sql noprint;
    select strip(put(max(_marker), best.)) into :mnum separated by ' '
    from figure;

    select strip(_marfclr), strip(_maroclr),
           strip(_marsz), strip(_marsbl), _marker
    into   :mfclr1 - :mfclr%left(&mnum), :moclr1 - :moclr%left(&mnum), :msz1 - :msz%left(&mnum), :msbl1 -
:msbl%left(&mnum), :dummy
    from figure
    where _marker>0
    order by _marker;
quit;

*** other macro variables for figure;
data _figure;
    set figure end=eof;
    by groupn group subgrpn subgrp;
    retain _totmax _totmin 0;
    if .<_totmax<ucl then _totmax=ucl;
    if .<lcl<_totmin then _totmin=lcl;
    if nmiss(_totmin, _totmax)=0 then _diff8=(_totmax-_totmin)/12;
    if .<_diff8=<0.01 then _figinc=0.01;
        else if 0.01<_diff8<=0.02 then _figinc=0.02;
        else if 0.02<_diff8<=0.05 then _figinc=0.05;
        else if 0.05<_diff8<=0.1 then _figinc=0.1;
        else if 0.1<_diff8<=0.2 then _figinc=0.2;
        else if 0.2<_diff8<=0.5 then _figinc=0.5;
        else if 0.5<_diff8<=1 then _figinc=1;
        else if 1<_diff8<=2 then _figinc=2;
```

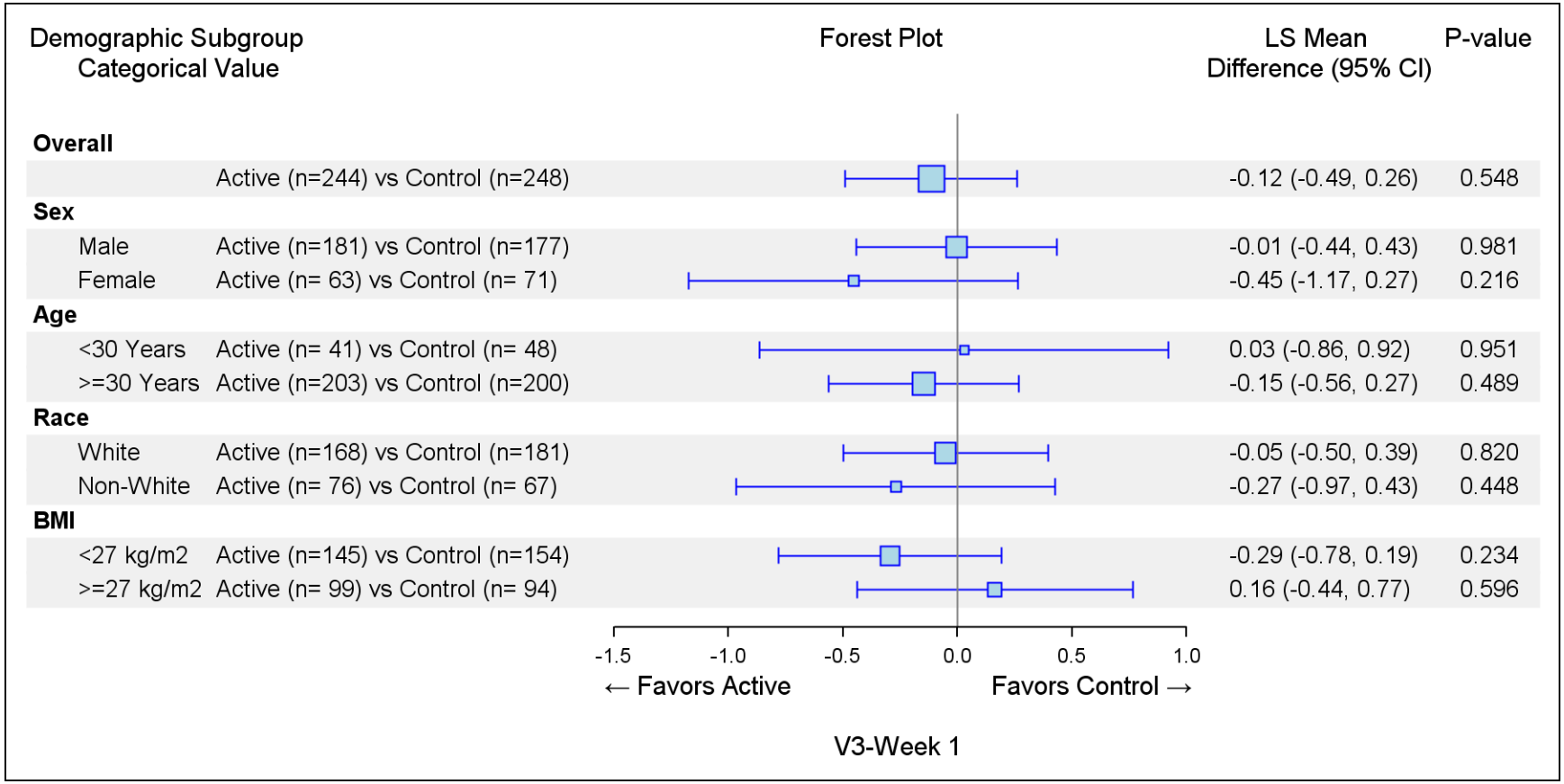
```
        else if 2<_diff8<=5 then _figinc=5;
        else if 5<_diff8<=10 then _figinc=10;
        else if 10<_diff8<=15 then _figinc=15;
        else if 15<_diff8<=20 then _figinc=20;
        else if 20<_diff8<=50 then _figinc=50;
        else if 50<_diff8<=100 then _figinc=100;
    _totmean=round(mean(_totmax, _totmin), _figinc);
    if eof then do;
        array ma{10} _max01-_max10;
        array mi{10} _min01-_min10;
        do j=1 to 10;
            if missing(_figinc)=0 then do;
                ma{j}=_totmean+_figinc*j;
                mi{j}=_totmean-_figinc*j;
            end;
            if ma{j}<_totmax then ma{j}=.;
            if mi{j}>_totmin then mi{j}=.;
        end;
        if nmiss(of _max01 - _max10)^=10 then _figmax=min(of _max01 - _max10);
        if nmiss(of _min01 - _min10)^=10 then _figmin=max(of _min01 - _min10);
        call symput('fmax', strip(put(_figmax, best8.)));
        call symput('fmin', strip(put(_figmin, best8.)));
        call symput('finc', strip(put(_figinc, best8.)));
    end;
run;

proc template;
    define style graphtmpl1;
        parent=styles.Appendix;
        class GraphFonts /
            'GraphDataFont'=("<sans-serif>, <MTsans-serif>",7pt)
            'GraphUnicodeFont'=("<MTsans-serif-unicode>",9pt)
            'GraphValueFont'=("<sans-serif>, <MTsans-serif>",9pt)
            'GraphLabelFont'=("<sans-serif>, <MTsans-serif>",10pt)
            'GraphFootnoteFont'=("<sans-serif>, <MTsans-serif>",9pt)
            'GraphTitleFont'=("<sans-serif>, <MTsans-serif>",11pt,bold)
            'GraphAnnoFont'=("<sans-serif>, <MTsans-serif>",10pt);
        class GraphData1 /
            markersymbol='squarefilled'
            linestyle=4
            contrastcolor=green
```

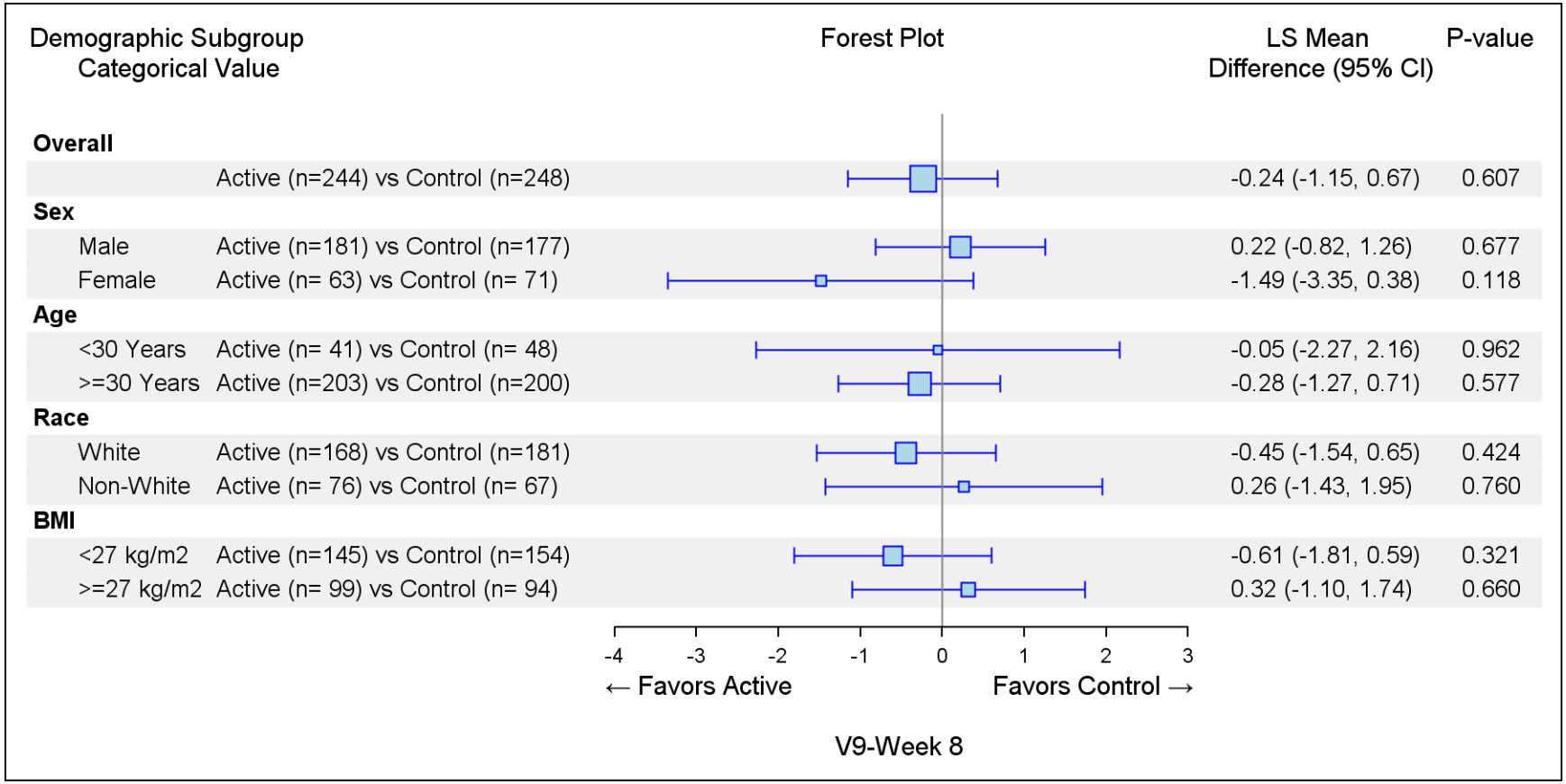
```
            color=green;
        class GraphData2 /
            markersymbol='circlefilled'
            linestyle=2
            contrastcolor=red
            color=red;
        class GraphData3 /
            markersymbol='diamondfilled'
            linestyle=1
            contrastcolor=blue
            color=blue;
    end;
run;
options orientation=landscape mprint mlogic;
%macro forest();
ods listing close;
ods rtf file="forest plot.rtf" style=graphtmpl1;
proc template;
    define statgraph forest;
        begingraph / designwidth=9in designheight=4.5in border=yes;
        discreteattrmap name='text' / ignorecase=true trimleading=true;
            value '1' / textattrs=(color=black weight=bold size=10);
            value '2' / textattrs=(color=black weight=normal  size=9);
        enddiscreteattrmap;
        discreteattrvar attrvar=grp var=_id attrmap='text';
        layout lattice / columns=1;
        *** Column headers;
        sidebar / align=top;
            layout lattice / rows=3 columns=4 columnweights=(0.35 0.42 0.15 0.07);
                entry textattrs=(size=10 weight=normal color=black) halign=left "&hl_lbl1" halign=right "
";
                entry textattrs=(size=10 weight=normal color=black) "&hc_lbl1";
                entry textattrs=(size=10 weight=normal color=black) "&hr_lbl1";
                entry textattrs=(size=10 weight=normal color=black) halign=right "&hr2_lbl1";

                entry textattrs=(size=10 weight=normal color=black) halign=left "&hl_lbl2" halign=right "
";
                entry textattrs=(size=10 weight=normal color=black) "&hc_lbl2";
                entry textattrs=(size=10 weight=normal color=black) "&hr_lbl2";
                entry textattrs=(size=10 weight=normal color=black) halign=right "&hr2_lbl2";
            endLayout;
```

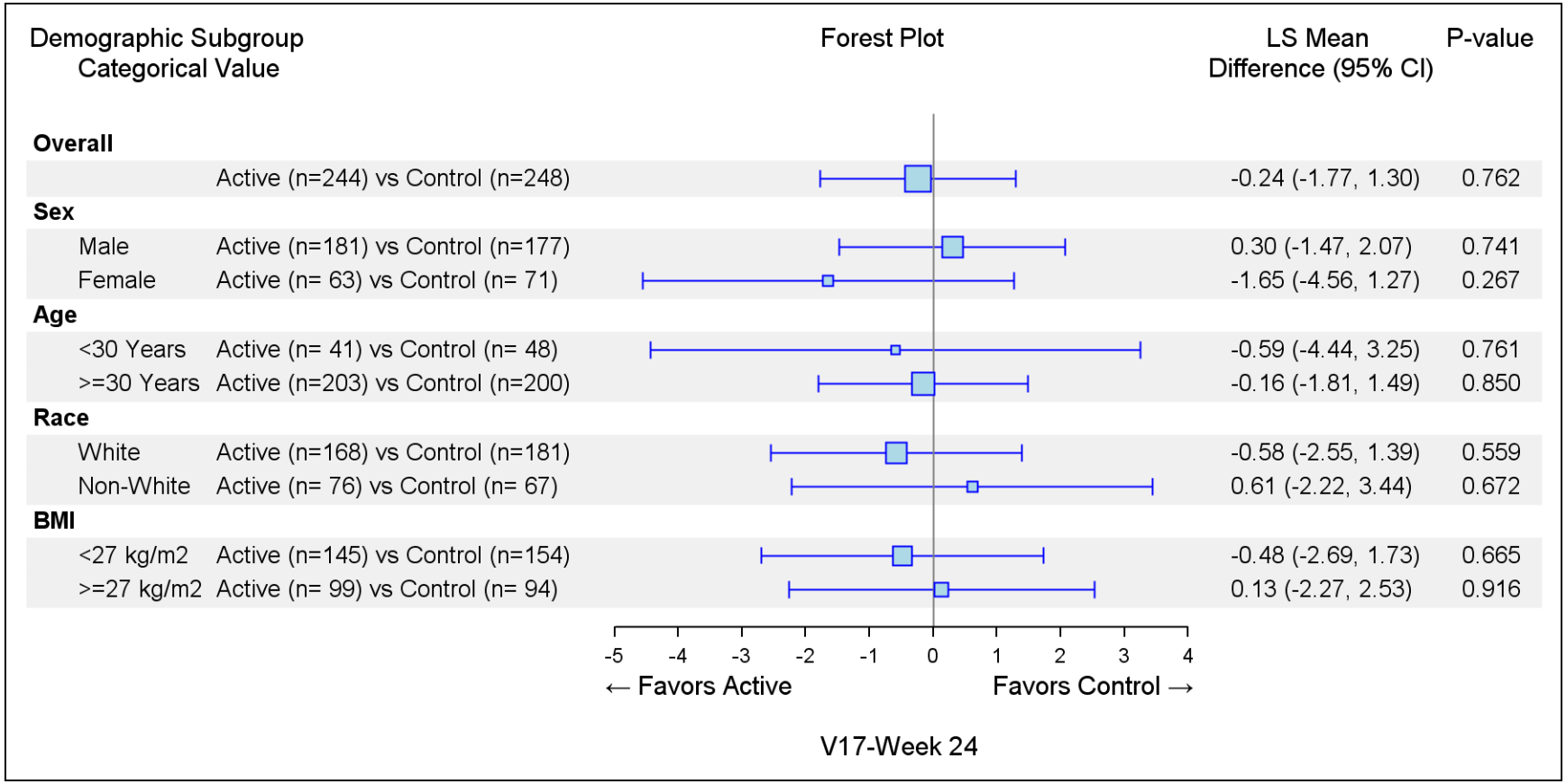
```
endsidebar;
layout overlay /
    xaxisopts=(display=(line ticks tickvalues) labelattrs=(size=9pt) tickvalueattrs=(size=8pt)
               linearopts=(tickvaluesequence=(start=&fmin end=&fmax increment=&finc)
                           viewmin=&fmin viewmax=&fmax) lineExtent=data)
    yaxisopts=(reverse=true display=none offsetmin=0 offsetmin=0.06) wallDisplay=none;
    *** left;
    innermargin / align=left;
        axistable y=_obs value=subgrp /
            indentweight=_indent display=(values) textgroup=grp valueattrs=(size=9);
        axistable y=_obs value=textvs /
            display=(values) showmissing=false valueattrs=(size=9);
    endinnermargin;
    *** center;
    referenceline y=_ref / lineattrs=(thickness=20 color=cxf0f0f0);
    highlowplot y=_obs low=lcl high=ucl /
        endcapdisplaypolicy=always highcap=serif lowcap=serif lineattrs=graphdata3;
    referenceline x=0;
    %do k=1 %to &mnum;
        scatterplot y=_obs x=eval(ifn(_marker=&k,_mean,.)) /
            filledoutlinedmarkers=true markerfillattrs=(color=&&mfclr&k)
            markeroutlineattrs=(color=&&moclr&k) markerattrs=(size=&&msz&k symbol=&&msbl&k);
    %end;
    *** right;
    innermargin / align=right;
        axistable y=_obs value=ci / display=(values) showmissing=false valueattrs=(size=9);
        axistable y=_obs value=_blank / display=(values) showmissing=false valueattrs=(size=9);
        axistable y=_obs value=_blank / display=(values) showmissing=false valueattrs=(size=9);
        axistable y=_obs value=pval / display=(values) showmissing=false valueattrs=(size=9);
        axistable y=_obs value=_blank / display=(values) showmissing=false valueattrs=(size=9);
    endinnermargin;
endlayout;
sidebar / align=bottom;
    layout lattice / columns=3 columnweights=(0.38 0.395 0.225);
        entry " ";
        entry textattrs=(size=10 weight=normal)
            halign=left "(*ESC*){unicode '2190'x} Favors Active" halign=right "Favors Control
                (*ESC*){unicode '2192'x}";
        entry " ";
    endLayout;
endsidebar;
```

```
        endlayout;
        endgraph;
    end;
run;

ods graphics / outputfmt=png;
proc sgrender data=figure template=forest;
run;

ods rtf close;
ods listing;
%mend forest;
%forest;
```

APPENDIX 2 FOREST PLOT OUTPUTS FOR ALL VISITS




Demographic Subgroup
Categorical Value
Forest Plot
LS Mean
Difference (95% CI)
P-value
Overall
Active (n=244) vs Control (n=248)
-0.24 (-1.15, 0.67)
0.607
Sex
Male
Active (n=181) vs Control (n=177)
0.22 (-0.82, 1.26)
0.677
Female
Active (n= 63) vs Control (n= 71)
-1.49 (-3.35, 0.38)
0.118
Age
<30 Years
Active (n= 41) vs Control (n= 48)
-0.05 (-2.27, 2.16)
0.962
>=30 Years
Active (n=203) vs Control (n=200)
-0.28 (-1.27, 0.71)
0.577
Race
White
Active (n=168) vs Control (n=181)
-0.45 (-1.54, 0.65)
0.424
Non-White
Active (n= 76) vs Control (n= 67)
0.26 (-1.43, 1.95)
0.760
BMI
<27 kg/m2
Active (n=145) vs Control (n=154)
-0.61 (-1.81, 0.59)
0.321
>=27 kg/m2
Active (n= 99) vs Control (n= 94)
0.32 (-1.10, 1.74)
0.660
-4
-3
-2
-1
0
1
2
3
← Favors Active
Favors Control →
V9-Week 8


Demographic Subgroup
Categorical Value
Forest Plot
LS Mean
Difference (95% CI)
P-value
Overall
Active (n=244) vs Control (n=248)
-0.24 (-1.77, 1.30)
0.762
Sex
Male
Active (n=181) vs Control (n=177)
0.30 (-1.47, 2.07)
0.741
Female
Active (n= 63) vs Control (n= 71)
-1.65 (-4.56, 1.27)
0.267
Age
<30 Years
Active (n= 41) vs Control (n= 48)
-0.59 (-4.44, 3.25)
0.761
>=30 Years
Active (n=203) vs Control (n=200)
-0.16 (-1.81, 1.49)
0.850
Race
White
Active (n=168) vs Control (n=181)
-0.58 (-2.55, 1.39)
0.559
Non-White
Active (n= 76) vs Control (n= 67)
0.61 (-2.22, 3.44)
0.672
BMI
<27 kg/m2
Active (n=145) vs Control (n=154)
-0.48 (-2.69, 1.73)
0.665
>=27 kg/m2
Active (n= 99) vs Control (n= 94)
0.13 (-2.27, 2.53)
0.916
-5 -4 -3 -2 -1 0 1 2 3 4
← Favors Active
Favors Control →
V17-Week 24