PharmaSUG 2020 - Paper AP-005

%MVMODELS: a Macro for Survival and Logistic Analysis

Jeffrey Meyers, Mayo Clinic

ABSTRACT

The research field of clinical oncology heavily relies on the methods of survival analysis and logistic regression. Analyses involve one or more variables within a model, and multiple models are often compared within subgroups. Results are prominently displayed within either a table or graphically with a forest plot. The MVMODELS macro performs every step for a univariate or multivariate analysis: running the analysis, organizing the results into datasets for printing or plotting, and creating the final output as a table or graph. MVMODELS is capable of running and extracting statistics from multiple models at once, performing subgroup analyses, outputting to most file formats, and contains a large variety of options to customize the final output. The macro MVMODELS is a powerful tool for analyzing and visualizing one or more statistical models.

INTRODUCTION

Clinical oncology trial endpoints heavily rely on survival or logistic regression analyses to determine whether the trial is positive or negative. These can include endpoints such as overall survival, progression-free survival, and confirmed tumor response status. The analyses are potentially performed within multiple different populations such as protocol defined stratification or descriptive factors. Models are either univariate (consisting of one covariate) or adjusted for other relevant factors in a multivariate model. The macro MVMODELS is a tool designed to handle all of these situations and output the results into either a clean and easy to read table or forest plot. The macro performs the analysis, organizes and combines the results, and outputs the final product all from one macro call. The MVMODELS macro is a powerful tool for any programmer that analyzes clinical trial data.

SAMPLE DATASET FOR EXAMPLES

The dataset used in the examples within this paper is randomly generated from the following code:

```
data random;
    call streaminit(123);
    array u {50};
    do study = 1 to 5;*Studies;
        do i = 1 to 500+floor(rand("Uniform")*500);*Patients;
            do j = 1 to dim(u);*Variables;
                u(j)=rand("Uniform");
                end;
            arm=catx(' ','Arm',1+round(u1,1));
            age=floor(18+62*u2);
            gender=ifc(u3>=0.5,'Male','Female');
            tstage=cats('T',1+floor(4*u4));
            nstage=cats('N',0+floor(3*u5));
            mstage=cats('M',0+floor(2*u6));

            if arm='Arm 1' then
                response=ifc(u7>0.5,'Response','No Response');
            else if arm='Arm 2' then
                response=ifc(u7>0.7,'Response','No Response');
            **Follow up to 10 years;
            os_time=ifn(arm='Arm 1',1+floor(120*u13),1+floor(120*u14));
            os_stat=ifn(arm='Arm 1',
                ifn((os_time <=60 and u8>0.35) or
                    (os_time > 60 and u8>0.65),1+floor(3*u9),0),
                ifn((os_time <=60 and u10>0.7) or
                    (os_time > 60 and u11>0.2),1+floor(u12*3),0));
            output;
            end;
        end;
    drop u: i j;
    label study='Study Number' arm='Treatment Arm' age='Age'
          gender='Gender' tstage='T-Stage' nstage='N-Stage'
        mstage='M-Stage' response='Response Status'
        os_time='Overall Survival Time (months)'
        os_stat='Overall Survival Status';
run;
```

The randomly generated dataset is not realistic clinical trial data, but will serve the purpose for the examples in this paper. This data is meant to represent a pooled analysis of five trials that all have the same two treatments (Arm 1 vs Arm 2). Arm 1 represents a treatment that is very aggressive early but is harder for the patient to tolerate, and Arm 2 represents a treatment that is easier on the patient but has less overall efficacy. The dataset contains demographic and disease characteristics for use in survival and logistic modeling. The status variable for overall survival has three different events that can happen for competing risks analysis.

EXAMPLE OUTPUT

The MVMODELS macro performs analysis and outputs the results into either a forest plot or a table. The macro parameters *SHOW_TABLE* and *SHOW_PLOT* determine which output is created. The following examples show the flexibility of both the types of analyses that can be performed and the ways the results can be displayed.

DISPLAYING RESULTS FROM A MULTIVARIATE MODEL

A multivariate model requires the flexibility to display different types of covariates in an easy to read format. The MVMODELS macro has multiple ways to display discrete or continuous covariates to best display the results in a meaningful format. The following code is an example of running a multivariate model:

```
%mvmodels(DATA=random, METHOD=survival, TIME=os_time, CENS=os_stat,
  CEN_VL=0, COVARIATES=arm age gender, TYPE=2 1 2, CAT_DISPLAY=4,
  CONT_STEP=10);
```

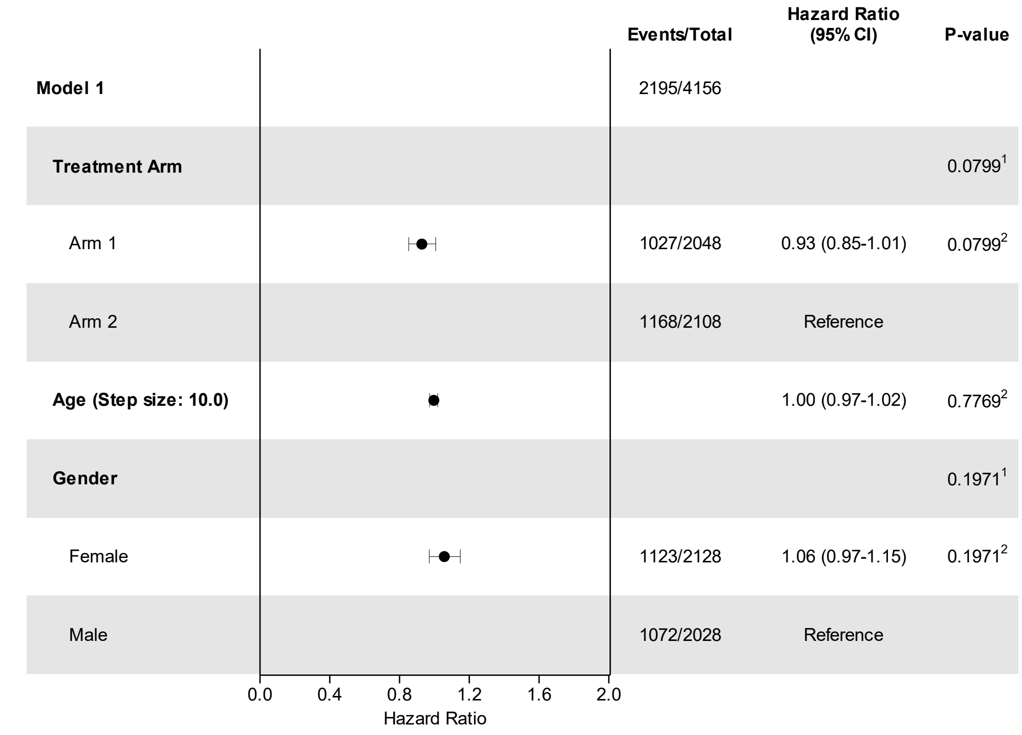
Table 1. Displays the table from a survival based multivariate model of overall survival based on treatment arm, age, and gender

	Events/Total	**Hazard Ratio (95% CI)**	**P-value**
Model 1	2195/4156		
Treatment Arm			0.0799[1]
Arm 1	1027/2048	0.93 (0.85-1.01)	0.0799[2]
Arm 2	1168/2108	Reference	
Age (Step size: 10.0)		1.00 (0.97-1.02)	0.7769[2]
Gender			0.1971[1]
Female	1123/2128	1.06 (0.97-1.15)	0.1971[2]
Male	1072/2028	Reference	

[1]Type 3 Wald p-value;[2]Covariate Wald p-value;

Table 1. The table follows the same structure as the plot. Alternating row shading is the default. P-value footnotes are automatically created. *CONT_STEP* changes the units of the continuous variable. Type 1 variables are continuous and type 2 variables are categorical.

Figure 1. Displays the forest plot from a survival based multivariate model of overall survival based on treatment arm, age, and gender



	Events/Total	Hazard Ratio (95% CI)	P-value
Model 1	2195/4156		
Treatment Arm			0.0799[1]
Arm 1	1027/2048	0.93 (0.85-1.01)	0.0799[2]
Arm 2	1168/2108	Reference	
Age (Step size: 10.0)		1.00 (0.97-1.02)	0.7769[2]
Gender			0.1971[1]
Female	1123/2128	1.06 (0.97-1.15)	0.1971[2]
Male	1072/2028	Reference	

[1]Type 3 Wald p-value; [2]Covariate Wald p-value;

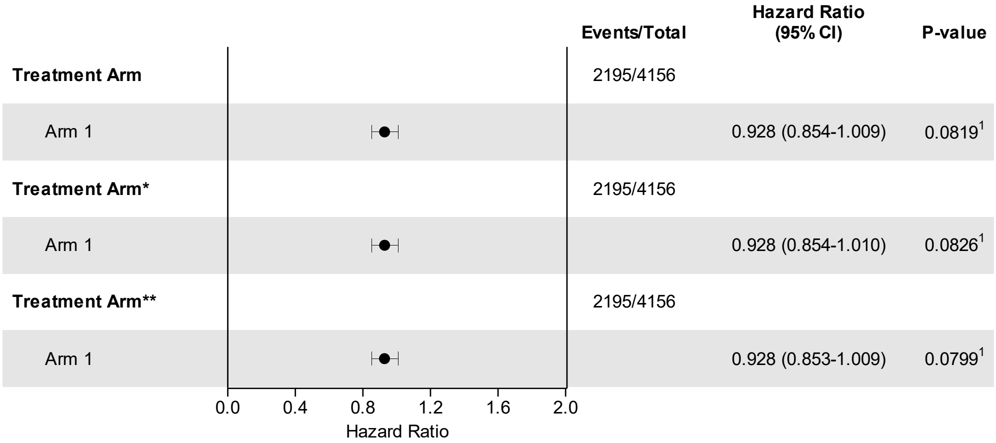
Figure 1. *CAT_DISPLAY*=4 will display all levels of the categorical covariate including the reference value.

COMPARING RESULTS FROM MULTIPLE MODELS

Different models are often compared within clinical trials to check the impact of adding one or more adjusting covariates. The MVMODELS macro can run more than one model at a time and can limit the display to one covariate of interest without showing the adjusting factors. This allows an easier comparison of the covariate of interest and a more compact plot or table. The number of models run is controlled by the *NMODELS* macro option, and different options can be specified for each model by using the pipe symbol as a delimiter (see *MODEL_TITLE* below). Options without the pipe delimiter are applied to all models (see *TIME* option below). The following code is an example of running multiple models:

```
%mvmodels(DATA=random, METHOD=survival, TIME=os_time, CENS=os_stat, CEN_VL=0,
   NMODELS=3, SHOW_ADJCOVARIATES=0,
   MODEL_TITLE=Treatment Arm|Treatment Arm*|Treatment Arm**,
   FOOTNOTE=*Adjusted for age; **Adjusted for age and gender, HRDIGITS=3,
   COVARIATES=arm |arm age |arm age gender, TYPE=2 1 2, CAT_DISPLAY=3,
   PVAL_TYPE3=0, HEIGHT=4in);
```

Figure 2. Displays the results from three different models.



	Events/Total	Hazard Ratio (95% CI)	P-value
Treatment Arm	2195/4156		
Arm 1		0.928 (0.854-1.009)	0.0819[1]
Treatment Arm*	2195/4156		
Arm 1		0.928 (0.854-1.010)	0.0826[1]
Treatment Arm**	2195/4156		
Arm 1		0.928 (0.853-1.009)	0.0799[1]

[1]Covariate Wald p-value;

*Adjusted for age; **Adjusted for age and gender

Figure 2. The first model is a univariate model of treatment. The second model adjusts for age, and the third model adjusts for age and gender. *CAT_DISPLAY=3* will display the current covariate value without the reference group.

Table 2. Displays the results from three different models.

	Events/Total	Hazard Ratio (95% CI)	P-value
Treatment Arm	2195/4156		
Arm 1		0.928 (0.854-1.009)	0.0819[1]
Treatment Arm*	2195/4156		
Arm 1		0.928 (0.854-1.010)	0.0826[1]
Treatment Arm**	2195/4156		
Arm 1		0.928 (0.853-1.009)	0.0799[1]

[1]Covariate Wald p-value;

*Adjusted for age; **Adjusted for age and gender

Table 2. Setting SHOW_ADJCOVARIATES=0 prevents adjusting factors from being displayed allowing the adjusted treatment arm covariate to be compared more easily.

QUICKLY PERFORMING SUBGROUP ANALYSES

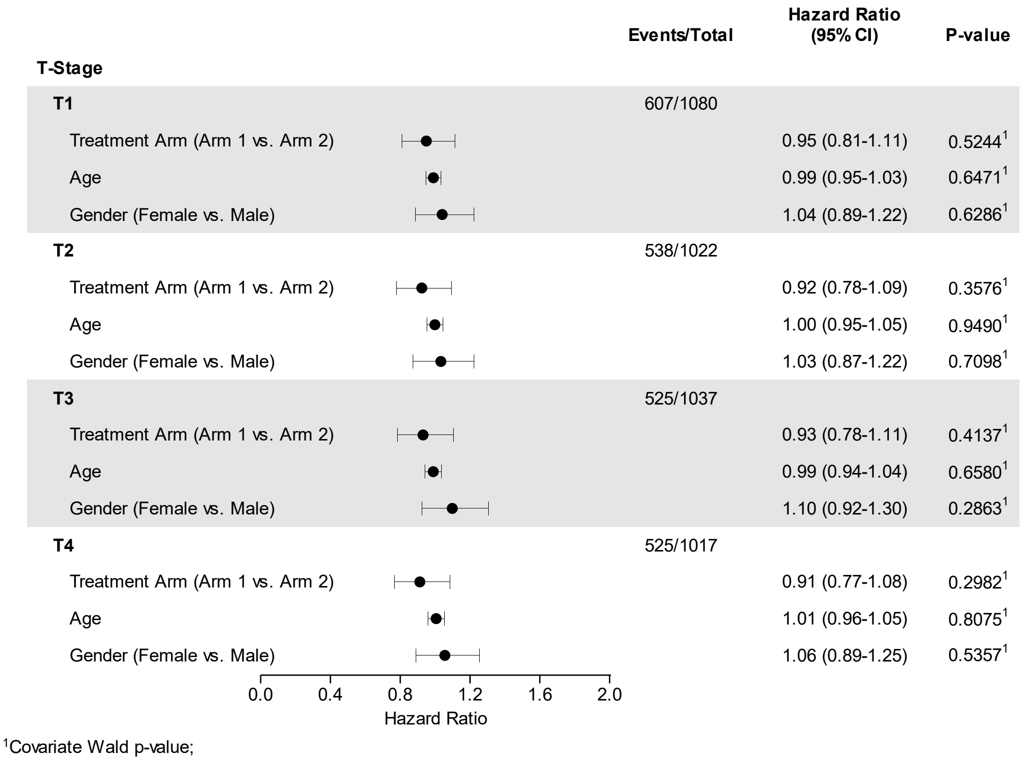
Subgroup analyses are very common in oncology research and meta-analyses. A subgroup analysis involves running the same model within the levels of another variable. An example of this would be comparing treatment arms within different genders. The MVMODELS macro has several options to easily display subgroup analyses in different ways.

Within the same cell (*BY* parameter)

The *BY* parameter allows for one or more variables to be listed. The same model will be run within each level of each variable specified. The order of the *BY* variable values can be changed with *BYORDER* parameter. The following example shows the use of the *BY* variable:

```
%mvmodels(DATA=random, METHOD=survival, TIME=os_time, CENS=os_stat, CEN_VL=0,
  COVARIATES=arm age gender, TYPE=2 1 2, CAT_DISPLAY=1, CONT_DISPLAY=2,
  CONT_STEP=10, BOLD_COV_LABEL=0, BY=tstage, SHADING=2, SHOWWALLS=0);
```

Figure 3. The Cox model of overall survival and treatment arm, age, and gender is computed within each level of T-stage.



	Events/Total	Hazard Ratio (95% CI)	P-value
T-Stage			
T1	607/1080		
Treatment Arm (Arm 1 vs. Arm 2)		0.95 (0.81-1.11)	0.5244[1]
Age		0.99 (0.95-1.03)	0.6471[1]
Gender (Female vs. Male)		1.04 (0.89-1.22)	0.6286[1]
T2	538/1022		
Treatment Arm (Arm 1 vs. Arm 2)		0.92 (0.78-1.09)	0.3576[1]
Age		1.00 (0.95-1.05)	0.9490[1]
Gender (Female vs. Male)		1.03 (0.87-1.22)	0.7098[1]
T3	525/1037		
Treatment Arm (Arm 1 vs. Arm 2)		0.93 (0.78-1.11)	0.4137[1]
Age		0.99 (0.94-1.04)	0.6580[1]
Gender (Female vs. Male)		1.10 (0.92-1.30)	0.2863[1]
T4	525/1017		
Treatment Arm (Arm 1 vs. Arm 2)		0.91 (0.77-1.08)	0.2982[1]
Age		1.01 (0.96-1.05)	0.8075[1]
Gender (Female vs. Male)		1.06 (0.89-1.25)	0.5357[1]

[1]Covariate Wald p-value;

Figure 3. *SHADING*=2 alternates the shading between *BY* levels to make it easier to visually distinguish groups. *CAT_DISPLAY*=1 combines two level covariates into one row to save space. *CONT_DISPLAY*=2 hides the step size text of the continuous variable label. *SHOW_WALLS*=0 removes the lines bordering the plot area.

Table 3. The Cox model of overall survival and treatment arm, age, and gender is computed within each level of T-stage.

	Events/Total	Hazard Ratio (95% CI)	P-value
T-Stage			
T1	607/1080		
Treatment Arm (Arm 1 vs. Arm 2)		0.95 (0.81-1.11)	0.5244[1]
Age		0.99 (0.95-1.03)	0.6471[1]
Gender (Female vs. Male)		1.04 (0.89-1.22)	0.6286[1]
T2	538/1022		
Treatment Arm (Arm 1 vs. Arm 2)		0.92 (0.78-1.09)	0.3576[1]
Age		1.00 (0.95-1.05)	0.9490[1]
Gender (Female vs. Male)		1.03 (0.87-1.22)	0.7098[1]
T3	525/1037		
Treatment Arm (Arm 1 vs. Arm 2)		0.93 (0.78-1.11)	0.4137[1]
Age		0.99 (0.94-1.04)	0.6580[1]
Gender (Female vs. Male)		1.10 (0.92-1.30)	0.2863[1]
T4	525/1017		

	Events/Total	Hazard Ratio (95% CI)	P-value
Treatment Arm (Arm 1 vs. Arm 2)		0.91 (0.77-1.08)	0.2982[1]
Age		1.01 (0.96-1.05)	0.8075[1]
Gender (Female vs. Male)		1.06 (0.89-1.25)	0.5357[1]

[1] Covariate Wald p-value;

Table 3. The different model outputs can easily be compared between different levels of T-stage.

By rows (*ROWBY*)

The ROW*BY* parameter allows for one variable to be listed. The same model will be run within each level of the variable specified. The order of the *ROWBY* variable values can be changed with *ROWBYORDER* parameter. *ROWBY* is different from *BY* in that it further separates the groups into distinct rows that can be separated with lines and adds vertical labels at the head of each row. The following example shows the use of the *ROWBY* variable:

```
%mvmodels(DATA=random, WHERE=study in(1 2 3), METHOD=survival, TIME=os_time,
  CENS=os_stat, CEN_VL=0, COVARIATES=arm age gender, TYPE=2 1 2,
  CAT_DISPLAY=2, CONT_DISPLAY=3,CONT_STEP=10, ROWBY=study, SHOWWALLS=0,
  SHADING=0, REFLINE=1, PVAL_COVARIATES=0);
```

Figure 4. The Cox model of overall survival and treatment arm, age, and gender is computed within each level of study.

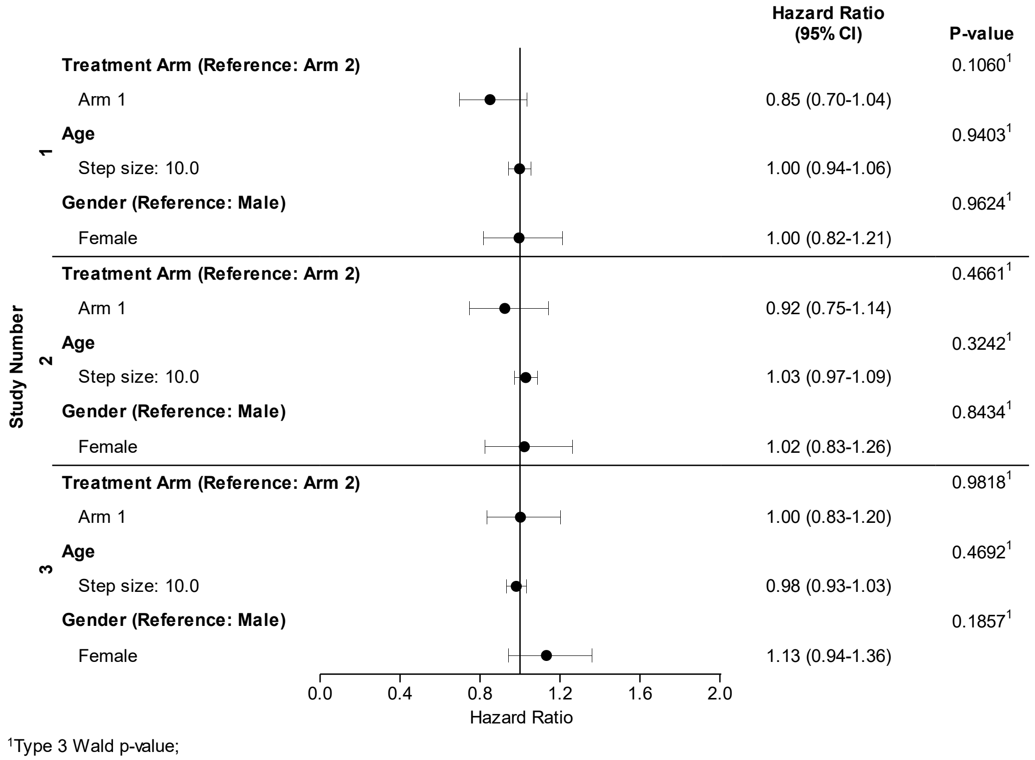


Figure 4. *SHADING*=0 removes the shading. *CAT_DISPLAY*=2 displays the reference group within the label. *CONT_DISPLAY*=3 moves the step size text to a new row. *REFLINE* adds a reference line to the graph to help visually compare estimates. *PVAL_COVARIATES*=0 disables the covariate level p-values.

Table 4. The Cox model of overall survival and treatment arm, age, and gender is computed within each level of study.

Study Number		Hazard Ratio (95% CI)	P-value
1	**Treatment Arm (Reference: Arm 2)**		0.1060[1]
	Arm 1	0.85 (0.70-1.04)	
	Age		0.9403[1]
	Step size: 10.0	1.00 (0.94-1.06)	
	Gender (Reference: Male)		0.9624[1]
	Female	1.00 (0.82-1.21)	
2	**Treatment Arm (Reference: Arm 2)**		0.4661[1]
	Arm 1	0.92 (0.75-1.14)	
	Age		0.3242[1]
	Step size: 10.0	1.03 (0.97-1.09)	
	Gender (Reference: Male)		0.8434[1]
	Female	1.02 (0.83-1.26)	
3	**Treatment Arm (Reference: Arm 2)**		0.9818[1]
	Arm 1	1.00 (0.83-1.20)	
	Age		0.4692[1]
	Step size: 10.0	0.98 (0.93-1.03)	
	Gender (Reference: Male)		0.1857[1]
	Female	1.13 (0.94-1.36)	

[1] Type 3 Wald p-value;

Table 4. *ROWBY* groups are easily distinguished with separating lines.

By columns (*COLBY*)

The *COLBY* parameter allows for one variable to be listed. The same model will be run within each level of the variable specified. The order of the *COLBY* variable values can be changed with *COLBYORDER* parameter. *COLBY* creates one column of summary statistics for each level of *COLBY*. The following example shows the use of the *COLBY* variable:

```
%mvmodels(DATA=random, METHOD=survival, TIME=os_time, CENS=os_stat, CEN_VL=0,
  COVARIATES=arm age gender, TYPE=2 1 2, CAT_DISPLAY=4, CONT_DISPLAY=2,
  CONT_STEP=10, COLBY=response, SHOWWALLS=0,
  UNDERLINEHEADERS=1, REFLINE=1, MIN=0, MAX=2, INCREMENT=0.5,
  PLOT_DISPLAY=subtitle ev_t hr_plot hr_est_range,
  PLOT_COLUMNWEIGHTS=0.2 0.2 0.3 0.3);
```

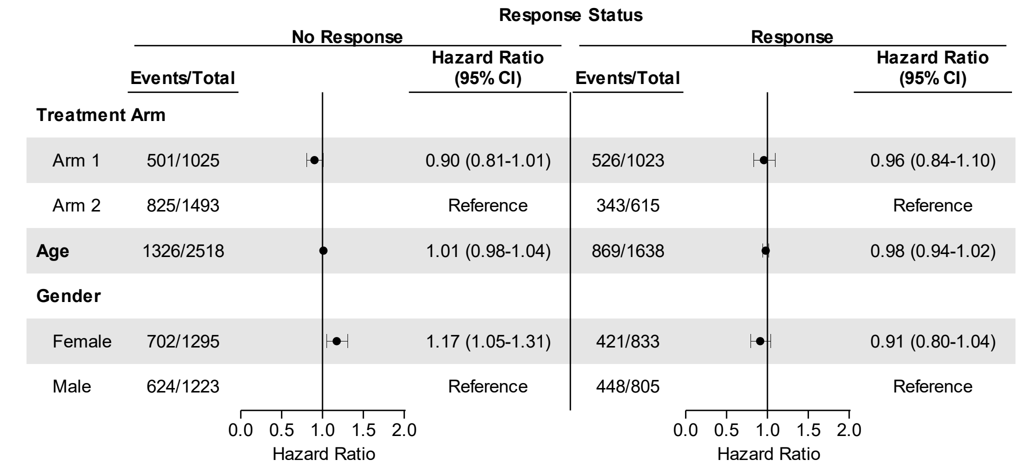
Table 5. The Cox model of overall survival and treatment arm, age, and gender is computed within each level of response.

	Response Status					
	No Response			Response		
	Events/Total	Hazard Ratio (95% CI)	P-value	Events/Total	Hazard Ratio (95% CI)	P-value
Treatment Arm			0.0677[1]			0.5265[1]
Arm 1	501/1025	0.90 (0.81-1.01)	0.0677[2]	526/1023	0.96 (0.84-1.10)	0.5265[2]
Arm 2	825/1493	Reference		343/615	Reference	
Age	1326/2518	1.01 (0.98-1.04)	0.4652[2]	869/1638	0.98 (0.94-1.02)	0.2670[2]
Gender			0.0037[1]			0.1714[1]
Female	702/1295	1.17 (1.05-1.31)	0.0037[2]	421/833	0.91 (0.80-1.04)	0.1714[2]
Male	624/1223	Reference		448/805	Reference	

[1]Type 3 Wald p-value;[2]Covariate Wald p-value;

Table 5. *PLOT_DISPLAY* only controls which summary stats are shown in the plot, while *TABLE_DISPLAY* determines which summary stats are shown in the table. A gap is added between each column.

Figure 5. The Cox model of overall survival and treatment arm, age, and gender is computed within each level of response.



	Response Status					
	No Response			Response		
	Events/Total		Hazard Ratio (95% CI)	Events/Total		Hazard Ratio (95% CI)
Treatment Arm						
Arm 1	501/1025		0.90 (0.81-1.01)	526/1023		0.96 (0.84-1.10)
Arm 2	825/1493		Reference	343/615		Reference
Age	1326/2518		1.01 (0.98-1.04)	869/1638		0.98 (0.94-1.02)
Gender						
Female	702/1295		1.17 (1.05-1.31)	421/833		0.91 (0.80-1.04)
Male	624/1223		Reference	448/805		Reference

[1]Type 3 Wald p-value; [2]Covariate Wald p-value;

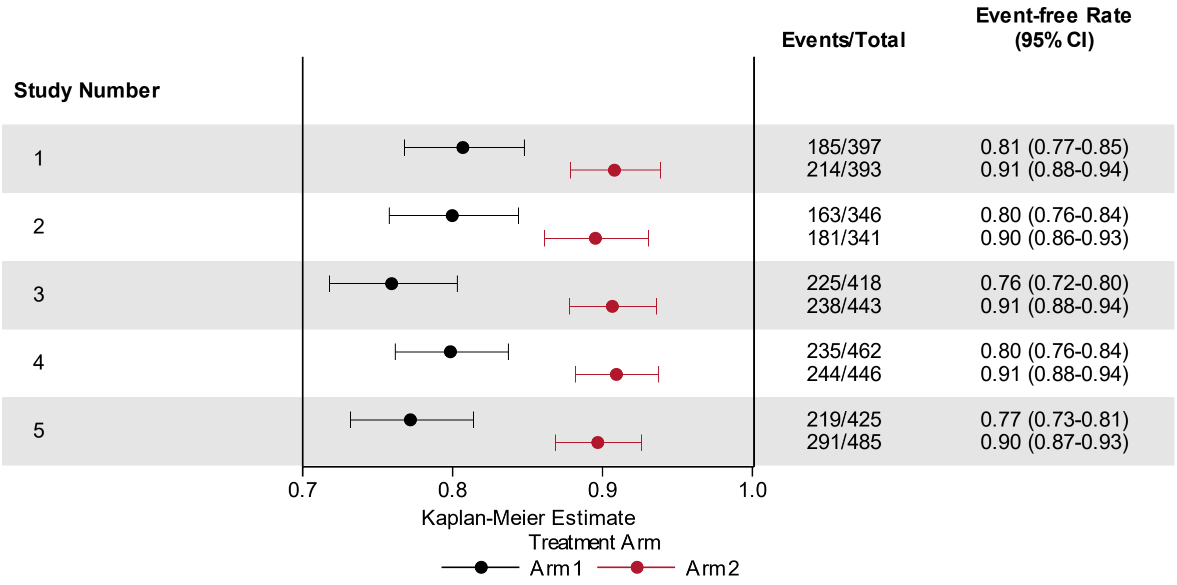
Figure 5. Each level of *COLBY* gets a column header and the *COLBY* label is shown at the top of the graph. *UNDERLINEHEADERS* underlines the headers of each column label. A vertical line separates each level of *COLBY*. The axes are set with the *MIN, MAX, and INCREMENT* covariates. *PLOT_DISPLAY* determines which summary statistics are shown, and *PLOT_COLUMNWEIGHTS* manually sets the space of each column.

Multiple results in same row (*GROUPBY*)

The *GROUPBY* parameter allows for one variable to be listed. The same model will be run within each level of the variable specified. The order of the *GROUPBY* variable values can be changed with *GROUPBYORDER* parameter. *GROUPBY* is useful for creating a very compact graph to compare two or more subgroups side-by-side. This is useful for case-control comparison graphs. The following example shows the use of the *GROUPBY* variable:

```
%mvmodels(DATA=random, METHOD=survival, TIME=os_time, CENS=os_stat, CEN_VL=0,
  BY=study, TIMELIST=36, GROUPBY=arm, MIN=0.70, MAX=1, INCREMENT=0.1);
```

Figure 6. Displays the 36 month Kaplan-Meier event-free rate for overall survival across each study grouped by arm.



Study Number	Events/Total	Event-free Rate (95% CI)
1	185/397 214/393	0.81 (0.77-0.85) 0.91 (0.88-0.94)
2	163/346 181/341	0.80 (0.76-0.84) 0.90 (0.86-0.93)
3	225/418 238/443	0.76 (0.72-0.80) 0.91 (0.88-0.94)
4	235/462 244/446	0.80 (0.76-0.84) 0.91 (0.88-0.94)
5	219/425 291/485	0.77 (0.73-0.81) 0.90 (0.87-0.93)

Figure 6. *TIMELIST* specifies one or more event-free time-points. *GROUPBY* displays the estimates in the same row but offset and in different colors. A legend is added to identify the different levels of *GROUPBY*.

Table 6. Displays the 36 month Kaplan-Meier event-free rate for overall survival across each study, grouped by arm.

	Treatment Arm	Events/Total	Event-free Rate (95% CI)
Study Number			
1			
	Arm 1	185/397	0.81 (0.77-0.85)
	Arm 2	214/393	0.91 (0.88-0.94)
2			
	Arm 1	163/346	0.80 (0.76-0.84)
	Arm 2	181/341	0.90 (0.86-0.93)
3			
	Arm 1	225/418	0.76 (0.72-0.80)
	Arm 2	238/443	0.91 (0.88-0.94)
4			
	Arm 1	235/462	0.80 (0.76-0.84)
	Arm 2	244/446	0.91 (0.88-0.94)
5			
	Arm 1	219/425	0.77 (0.73-0.81)
	Arm 2	291/485	0.90 (0.87-0.93)

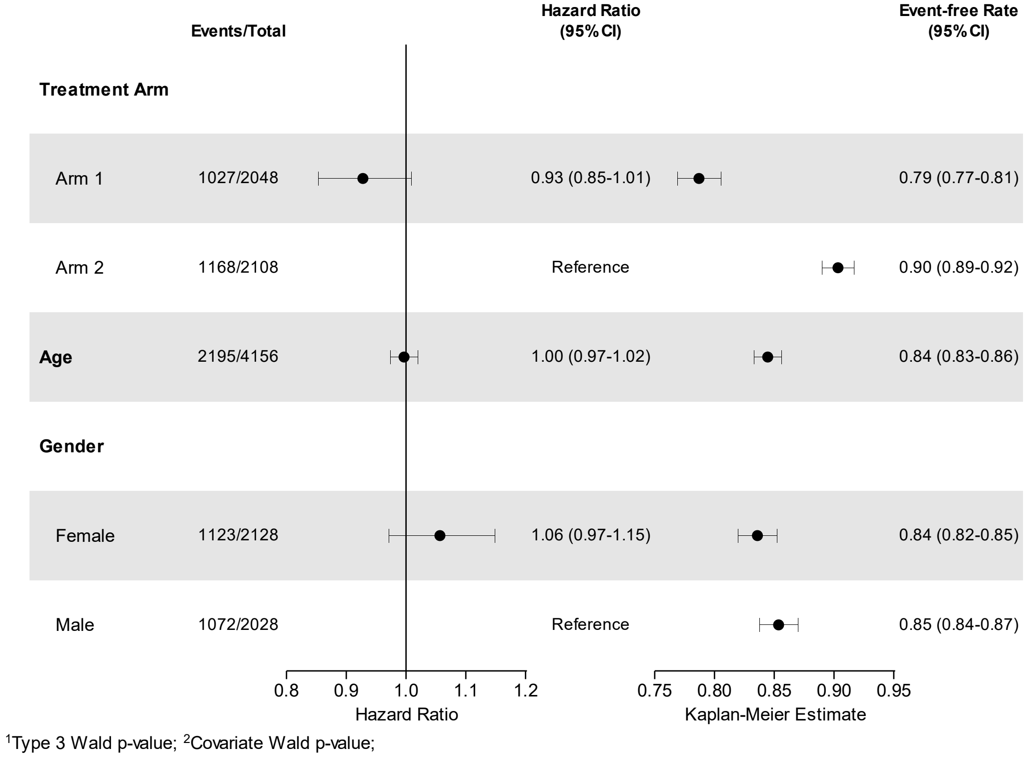
Table 6. *GROUPBY* adds a column to identify which level of *GROUPBY* the estimate belongs to. This option is more visually appealing in the graph, but is still available for the table.

DISPLAY MULTIPLE GRAPHS

The MVMODELS macro can display more than one graph within the same plot. The graphs must come from the same analysis. The following example shows having more than one graph within a plot:

```
%mvmodels(DATA=random, METHOD=survival, TIME=os_time, CENS=os_stat, CEN_VL=0,
  COVARIATES=arm age gender, TYPE=2 1 2, CAT_DISPLAY=4, CONT_DISPLAY=2,
  CONT_STEP=10, TIMELIST=36, SHOW_MODELSTATS=0, SHOWWALLS=0, MIN=0.8|0.75,
  MAX=1.2|0.95, INCREMENT=0.1|0.05, REFLINE=1|, SUMSIZE=9pt,
  PLOT_DISPLAY=subtitle ev_t hr_plot hr_est_range km_plot1 km_est_range1);
```

Figure 7. Graphs hazard ratios and 36 month overall survival event-free rates.



	Events/Total	Hazard Ratio (95% CI)	Event-free Rate (95% CI)
Treatment Arm			
Arm 1	1027/2048	0.93 (0.85-1.01)	0.79 (0.77-0.81)
Arm 2	1168/2108	Reference	0.90 (0.89-0.92)
Age	2195/4156	1.00 (0.97-1.02)	0.84 (0.83-0.86)
Gender			
Female	1123/2128	1.06 (0.97-1.15)	0.84 (0.82-0.85)
Male	1072/2028	Reference	0.85 (0.84-0.87)

Figure 7. Separate options such as *MIN* and *MAX* can be set for each graph. Each graph will have the _PLOT suffix within the *PLOT_DISPLAY* parameter.

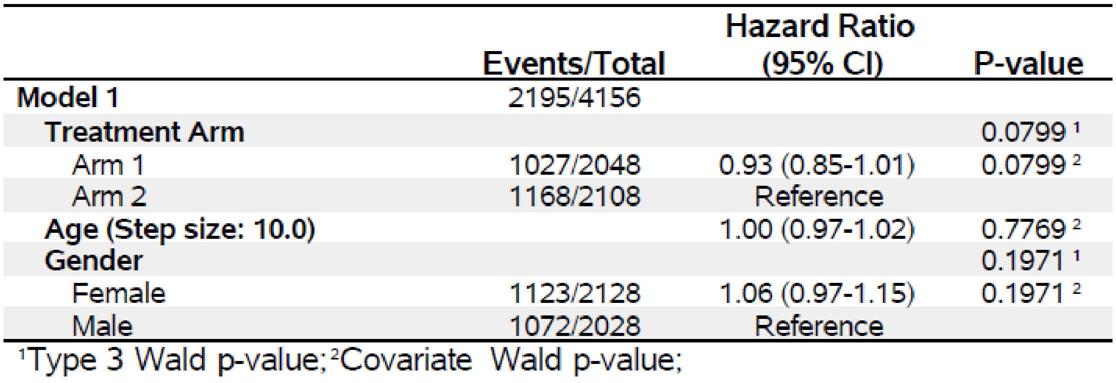
OUTPUT TO MULTIPLE DESTINATIONS

The MVMODELS macro is designed to output the plot and table to multiple destinations and have the same general appearance and style. The following is an example of outputting the table to multiple destinations at the same time:

```
ods pdf file='~/ibm/test.pdf' notoc bookmarkgen=no startpage=no;
ods excel file='~/ibm/test.xlsx' options (sheet_interval='none');
ods powerpoint file='~/ibm/test.pptx';
%mvmodels(DATA=random, METHOD=survival, TIME=os_time, CENS=os_stat, CEN_VL=0,
    COVARIATES=arm age gender, TYPE=2 1 2, CAT_DISPLAY=4, CONT_STEP=10);
ods _all_ close;
```

ODS PDF output

Figure 8. Screen shot of the table from the PDF output.



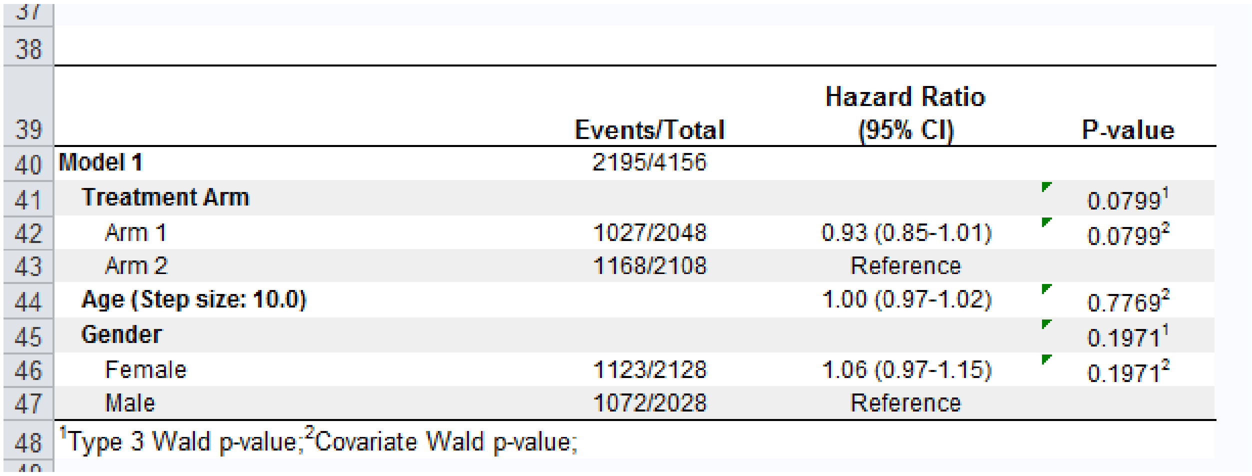
	Events/Total	Hazard Ratio (95% CI)	P-value
Model 1	2195/4156		
Treatment Arm			0.0799 [1]
Arm 1	1027/2048	0.93 (0.85-1.01)	0.0799 [2]
Arm 2	1168/2108	Reference	
Age (Step size: 10.0)		1.00 (0.97-1.02)	0.7769 [2]
Gender			0.1971 [1]
Female	1123/2128	1.06 (0.97-1.15)	0.1971 [2]
Male	1072/2028	Reference	

[1]Type 3 Wald p-value;[2]Covariate Wald p-value;

Figure 8. The space between the superscripts and p-values is due to the PDF destination

ODS EXCEL output

Figure 9. Screen shot of the table from the EXCEL output.



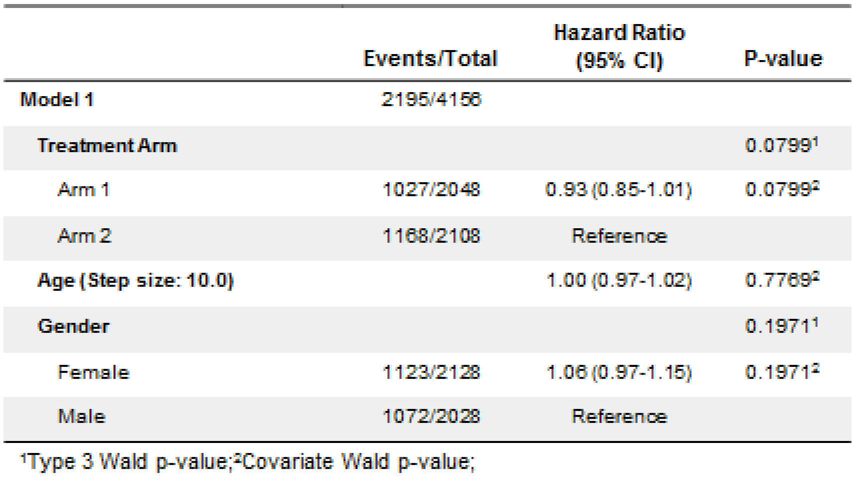
	Events/Total	Hazard Ratio (95% CI)	P-value
Model 1	2195/4156		
Treatment Arm			0.0799[1]
Arm 1	1027/2048	0.93 (0.85-1.01)	0.0799[2]
Arm 2	1168/2108	Reference	
Age (Step size: 10.0)		1.00 (0.97-1.02)	0.7769[2]
Gender			0.1971[1]
Female	1123/2128	1.06 (0.97-1.15)	0.1971[2]
Male	1072/2028	Reference	

[1]Type 3 Wald p-value;[2]Covariate Wald p-value;

Figure 9. The ability to output to EXCEL opens up the freedom of horizontal and vertical space for the table

ODS POWERPOINT output

Figure 10. Screen shot of the table from the POWERPOINT output.



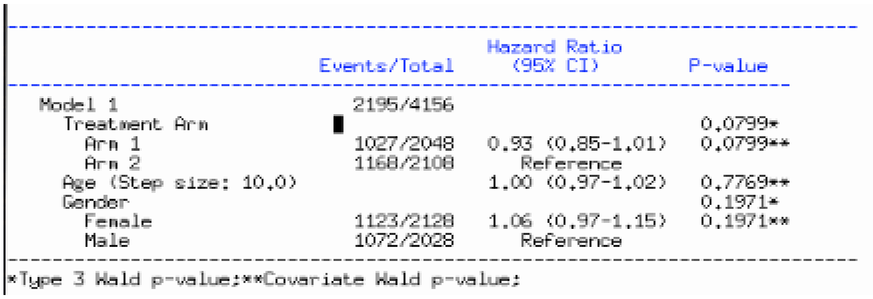
	Events/Total	Hazard Ratio (95% CI)	P-value
Model 1	2195/4156		
Treatment Arm			0.0799[1]
Arm 1	1027/2048	0.93 (0.85-1.01)	0.0799[2]
Arm 2	1168/2108	Reference	
Age (Step size: 10.0)		1.00 (0.97-1.02)	0.7769[2]
Gender			0.1971[1]
Female	1123/2128	1.06 (0.97-1.15)	0.1971[2]
Male	1072/2028	Reference	

[1]Type 3 Wald p-value;[2]Covariate Wald p-value;

Figure 10. The ability to output to POWERPOINT makes it easier to create summary slides from an analysis.

ODS LISTING output

Figure 11. Screen shot of the table from the LISTING output.



```
-------------------------------------------------------------------------------
                                              Hazard Ratio
                           Events/Total        (95% CI)          P-value
---------------------------------------------------------------------------
   Model 1                      2195/4156
     Treatment Arm                                                 0.0799*
       Arm 1                    1027/2048     0.93 (0.85-1.01)     0.0799**
       Arm 2                    1168/2108        Reference
     Age (Step size: 10.0)                    1.00 (0.97-1.02)     0.7769**
     Gender                                                        0.1971*
       Female                   1123/2128     1.06 (0.97-1.15)     0.1971**
       Male                     1072/2028        Reference
-------------------------------------------------------------------------------
*Type 3 Wald p-value;**Covariate Wald p-value;
```

Figure 11. The ability to output to the OUTPUT window with ODS LISTING allows for the results to be saved to the .LST file and makes for quick and compact results.

MACRO PROCESS OVERVIEW

STEP 1: ERROR CHECKING

The macro contains a large amount of parameters, so it is necessary to have error checking code throughout the macro to try to identify inappropriate macro parameter inputs before they cause errors in the SAS session. The error checking code makes sure variables exist, that required parameters are entered, and that proper values are entered. For instance, if a parameter has a designated list of values, the macro will check whether the user entered an appropriate value. If the user entered a value that does not match the list, then the macro stops, displays an error message, and provides the list of allowed values.

STEP 2: AUTOMATED ANALYSIS

Creating a forest plot that compares multiple models requires a large amount of code replication. Nearly identical code is repeated to run each model, and nearly identical code is used to extract and combine the results from each model. Programmers not familiar with writing macros will need to spend a great amount of time writing out many of these near duplicate sections of code to create one forest plot. The chance of a programming error also increases when duplicating the same code, especially if the code needs to be modified. The MVMODELS macro removes the time investment and risk by fully automating two types of analyses: survival analysis and logistic regression. Included within survival analysis are Kaplan-Meier event-free rates, median time-to-event, Cox proportional hazards ratios, and concordance index. Included with logistic regression are odds ratio, binomial success rates, and concordance indexes. Each of these analyses is customizable with macro parameters, and the detail of each method is listed in section 4.

STEP 3: PLOT DATA SET CONSTRUCTION

The macros generate a data set that is conducive to creating a forest plot or table.

STEP 4: GENERATE THE PLOT

The Graph Template Language (GTL) within the TEMPLATE procedure is used to set up the plot with a combination of the variables in the plot data set and macro variables derived from the plot data set. The actual image is then created using the SGRENDER procedure in combination with ODS Graphics option settings. The image can be a number of file types including PNG, EMF, PDF, JPEG, TIFF, and SVG, and can be embedded into RTF, HTML or PDF destinations.

STEP 4: GENERATE THE TABLE

The REPORT procedure is used to create the outputted table in combination with a large number of style modifications. The table is formatted to work well in RTF, HTML PDF, EXCEL, and POWERPOINT destinations.

ANALYSIS METHODS

The macro uses SAS procedures to perform the analyses. The following sections describe which SAS procedures are used to create each available statistic.

SURVIVAL METHODS

The MVMODELS macro can perform regular survival analysis as well as cumulative incidence analysis (SAS 9.4M3+). The methods to compute statistics differs slightly depending on which survival method is being used.

Survival and 1-Survival

- **Number of patients and events:** ODS OUTPUT statement within the LIFETEST procedure specifying the CENSOREDSUMMARY data set.
- **Kaplan-Meier event-free rates:** The TIMELIST option within the LIFETEST procedure is used to specify the time-points, and the OUTSURV option with the REDUCEOUT option is used to output the rates to a data set.
- **Median time-to-event:** ODS OUTPUT statement within the LIFETEST procedure specifying the QUARTILES data set. This data set is then subset down to where PERCENT=50.
- **P-values to compare survival curves (no stratification):** ODS OUTPUT specifying the HOMTESTS dataset. The *BY Variable* is listed in the STRATA statement and LOGRANK and WILCOXON are specified as options.
- **P-values to compare survival curves (with stratification):** ODS OUTPUT specifying the HOMTESTS dataset. Stratification factors are listed in the STRATA statement and LOGRANK, WILCOXON, and GROUP=*BY Variable* are specified as options.
- **Hazard ratios (stratified or not stratified):** ODS OUTPUT within the PHREG procedure specifying the PARAMETERESTIMATES data set. Continuous covariates are only included within the MODEL statement, and discrete covariates are included in both the MODEL and CLASS statements. The reference group for the categorical covariates is determined by the *CAT_REF* parameter and specified with the REF= option of the CLASS statement. The step size for continuous covariates is determined by the *CONT_STEP* parameter and is applied as a transformation to the hazard ratio after outputting using the following formula:

   - EXP(*CONT_STEP**LOG(HAZARDRATIO))

- **Cox model type-III tests (stratified or not stratified):** ODS OUTPUT within the PHREG procedure specifying the MODELANOVA data set. The TYPE3(ALL) option is specified within the MODEL statement in order to produce the score, Wald, and likelihood-ratio tests.
- **Cox model p-values comparing individual covariate levels (stratified or not stratified):** ODS OUTPUT within the PHREG procedure specifying the PARAMETERESTIMATES data set.
- **Concordance Indexes (stratified or not stratified):** The method for calculating concordance indexes described in the survConcordance[1] package from R developed by Therry Therneau is a widely recognized method, and thus was chosen as the method within this macro. The method uses a binary tree approach to calculating the weights, sum of squares, and eventual standard error. The model predicted values used in the binary tree method are taken from the OUTPUT statement within the PHREG procedure defining the XBETA variable.

Cumulative Incidence (CIF)

- **Number of patients and events:** ODS OUTPUT statement within the LIFETEST procedure specifying the FAILURESUMMARY data set. The event of interest is specified with the EVENTCODE option of the TIME statement.
- **Kaplan-Meier event-free rates:** The TIMELIST option within the LIFETEST procedure is used to specify the time-points, and the OUTCIF option is used to output the rates to a data set.
- **Median time-to-event:** The OUTCIF option within the LIFETEST procedure is used to output the cumulative incidence rates across time. A SQL procedure query is used pull the median times and confidence intervals from these rates. For example, the median time is the first time the CIF rate is greater than or equal to 50%.
- **P-values to compare survival curves (no stratification):** ODS OUTPUT specifying the GRAYTEST dataset. The *BY Variable* is listed in the STRATA statement.
- **P-values to compare survival curves (with stratification):** ODS OUTPUT specifying the GRAYTEST dataset. Stratification factors are listed in the STRATA statement and GROUP=*BY Variable* are specified as options.
- **Hazard ratios (stratified or not stratified):** ODS OUTPUT within the PHREG procedure specifying the PARAMETERESTIMATES data set. The event of interest is specified with the EVENTCODE option in the MODEL statement. Continuous covariates are only included within the MODEL statement, and discrete covariates are included in both the MODEL and CLASS statements. The reference group for the categorical covariates is determined by the *CAT_REF* parameter and specified with the REF= option of the CLASS statement. The step size for continuous covariates is determined by the *CONT_STEP* parameter and is applied as a transformation to the hazard ratio after outputting using the following formula:
   - EXP(*CONT_STEP**LOG(HAZARDRATIO))
- **Cox model type-III tests (stratified or not stratified):** ODS OUTPUT within the PHREG procedure specifying the MODELANOVA data set. The TYPE3(ALL) option is specified within the MODEL statement in order to produce the score, Wald, and likelihood-ratio tests.
- **Cox model p-values comparing individual covariate levels (stratified or not stratified):** ODS OUTPUT within the PHREG procedure specifying the PARAMETERESTIMATES data set.
- **Concordance Indexes (stratified or not stratified):** Not available with CIF method.

LOGISTIC REGRESSION METHODS

- **Number of patients and events / binomial success rates:** The TABLES statement is used with the BIN option to generate the binomial success rates along with number of patients and events. A success is considered to be the event of interest. A data step is utilized beforehand to create a variable that contains the counts for each level of the binomial variable. This variable is then used in a FREQ statement with the ZEROS option (which forces PROC FREQ to include counts of zero) to avoid errors that can arise with zero percent success rates and 100 percent success rates. The estimates are then output using an OUTPUT statement with the BIN option. The number of patients and events are output by using the OUT option within the TABLES statement. The counts and binomial success rates are taken within each level of a *BY variable* and within all levels of a categorical covariate.
- **P-values to compare *BY Variable* groups:** OUTPUT statement within the FREQ procedure with the CHISQ and/or FISHER options specified in the TABLES and OUTPUT statements as options.
- **Odds ratios (stratified and unstratified):** ODS OUTPUT within the LOGISTIC procedure specifying the ODDSRATIOSWALD data set. One ODDSRATIO statement is created for each covariate. The event is specified by the *EVENT* parameter. Continuous covariates are only included within the MODEL statement, and discrete covariates are included in both the MODEL and CLASS statements. The reference group for the categorical covariates is determined by the *CAT_REF* parameter and specified with the REF= option of the CLASS statement. The step size for continuous covariates is determined by the *CONT_STEP* parameter and is applied as a transformation to the odds ratio after outputting using the following formula:
    - EXP(*CONT_STEP**LOG(ODDSRATIO))
- **Logistic model type-III tests (stratified or not stratified):** ODS OUTPUT within the PHREG procedure specifying the MODELANOVA data set. The Wald test is created with this method.
- **Logistic model p-values comparing individual covariate levels (stratified or not stratified):** ODS OUTPUT within the PHREG procedure specifying the PARAMETERESTIMATES data set.
- **Concordance Indexes (not stratified):** The method for calculating concordance indexes follows the methods described in a paper by JA Hanley and BJ McNeil[2]. While the concordance index for logistic regression can be automatically output from the LOGISTIC procedure, the standard error is not. Without a standard error the confidence bounds for the concordance index cannot be calculated. The paper by Hanley and McNeil provide a method for calculating the standard error that has been commonly used within the Biomedical Statistics and Informatics division at Mayo Clinic. The model predicted values used in this method are taken from the OUTPUT statement defining the XBETA variable.

STREAMLINING THE ANALYSES

The MVMODELS macro has the potential to run a large number of statistical models in one macro call with the subgrouping *BY, COLBY, ROWBY,* and *GROUPBY* options, so the macro uses methods to streamline how many procedures need to be run to accommodate all of the models. The primary method used is to combine data set duplication with BY statements in the procedures. An example of this would be calculating the five year survival rates in the following example:

```
%mvmodels(DATA=random, METHOD=survival, TIME=os_time, CENS=os_stat, CEN_VL=0,
    TIMELIST=60, BY=arm gender tstage nstage);
```

There are eleven five year survival rate estimates that would need to be calculated from this macro call: two for ARM, two for GENDER, four for TSTAGE, and three for NSTAGE. This would normally require four calls to the LIFETEST procedure due to patients existing within multiple *BY* variables, but the macro bypasses this by duplicating each patient once for every *BY* variable level that they are in. The data set is set up with the following structure:

Table 7. Shows the duplicated data set structure

Patient ID	BY Variable	By Variable Level	Survival Time	Survival Status
1	ARM	1	30	0
2	ARM	2	60	1
1	GENDER	Female	30	0
2	GENDER	Male	60	1

Table 7. Each unique *BY* variable and level has its own subgroup within the transformed dataset. Patients exist in multiple subgroups.

Now the BY variable and BY variable level columns in table 7 can be used within the BY statement of the LIFETEST procedure creating all of the eleven estimates with only one LIFETEST procedure call instead of four. This method is also used when accommodating any *COLBY, ROWBY,* and *GROUPBY* variables as well as when calculating estimates for any categorical covariate levels. All models called by the macro are combined into a single dataset so that there is only one LIFETEST procedure called in the entire macro when calculating survival estimates, number of patients and events, and median time-to-event estimates.

This method is modified when running models with the PHREG or LOGISTIC procedures. The difficulty encountered is that multiple models can have varying numbers of covariates included and different stratification factors. The MVMODELS macro accounts for this in three steps:

1. Make a generic numeric variable for each categorical covariate where the numeric value is the order. For example gender would be converted from Male/Female to 1/2
2. Make the reference value of each categorical covariate equal to 0. This is so “0” can be specified as the reference group within the CLASS statement. For the gender example if males are the reference group then all of the male gender codes would be converted to zeros.
3. If one model has less covariates than another (e.g. model 1 uses arm and gender, model 2 uses arm, gender, and T-stage), then a variable is still created for the covariates that don’t exist in the current model, but all values are set to be 0. In the PHREG procedure a covariate that has all of the same values is thrown out of the model automatically.

A visual example of this method is shown in Table 8 below. Model one has Arm and Gender for covariates. Model two only has Arm for a covariate.

Table 8. Shows the data set structure for running the PHREG procedure

Model	Patient ID	Arm	Gender	Covariate 1	Covariate 2
1	1	1	Male	1	1
1	2	2	Female	0	0
2	1	1	Male	1	0
2	2	2	Female	0	0

Table 8. Model 1 uses Female and Arm 2 as reference groups, so the generic covariate variables are assigned values of 0. The other values are the numeric order of the remaining values. Model 2 does not use Gender as a covariate, so all values of the covariate 2 variable are 0.

All models are run with a single PHREG or LOGISTIC procedure call when using this methodology for the categorical covariates, continuous covariates, and strata variables along with the earlier method of duplicating the data set for each subgroup. One note is that all stratified and unstratified models are run separately in the LOGISTIC procedure as different results are created when using a stratification variable with all of the same levels versus not including the stratification variable completely.

OUTPUT METHODS

After analyzing the requested models the MVMODELS macro creates a dataset that combines the results together that is ideal for graphing and printing. The macro then builds a graph template with the TEMPLATE procedure, and prints the table with the REPORT procedure.

HOW THE DATASET IS CONSTRUCTED

The plot dataset is constructed using a combination of the SQL procedure and DATA steps. A DATA step is used to set up a structure based on input macro parameters for merging in the analysis results with the SQL procedure. The SQL procedure is used for merging because of the flexibility the procedure has with being able to use functions and other logic conditions in the merging process.

VARIABLE COMPONENTS

The variables that make up the plot data set are separated into different subtypes.

Row headers

Each row of the forest plot has a rowheader, or subtitle, to show the variable label or value. The row header is contained within one variable:

- SUBTITLE: Contains the row header such as variable label, variable level, group label, group level, or model title.

Estimates and Confidence Limits

There are numeric variables for the each calculated estimate, such as hazard ratio or odds ratio, and the upper and lower confidence limits. These variables are used for making the scatterplot and error bars in the plot.

- *PREFIX*_EST: Calculated ratio/estimate/rate
- *PREFIX*_LCL: Calculated lower 95% confidence limit
- *PREFIX*_UCL: Calculated upper 95% confidence limit

Multiple estimates are available to be plotted and displayed depending on the method used, so the *PREFIX_* term changes depending on which type of estimate is being captured. For example, the hazard ratios have the prefix HR_, odds ratios have the prefix OR_, and concordance indexes have the prefix C_. There are additional variables that are combinations of these variables. These are character variables with pre-specified rounding (set by macro parameters):

- *PREFIX*_ESTIMATE: ESTIMATE. Example: 1.34
- *PREFIX*_RANGE: LCL - UCL. Example: 0.25-4.20
- *PREFIX*_EST_RANGE: ESTIMATE (LCL - UCL). Example: 1.34 (0.25-4.20)

Number of Patients and Events

The section for number of patients and events has additional variables that can be used in the summary statistics panel with special formats to save space:

- TOTAL: Total number of patients within model/group
- EVENTS: Number of events or successes (for binomial) within model/group
- EV_T: Events/Total. Example: 245/300
- PCT: Percentage %. Example: 60%
- EV_T_PCT: Events/Total (Percentage %). Example: 245/300 (81.6%)

There are additional variables when utilizing Cox or logistic modeling that have the prefix REF_. These variables contain the number of patients and/or number of events of just the reference group.

P-Values

There are two variables allocated for the p-values.

- PVAL: character variable (applies PVALUE6.4 format to numeric value) that contains the p-value
- PFOOT_INDEX: contains the index of the p-value for the footnote. This is a numeric variable that will apply the value as a superscript if *AUTOPFOOT*=1.

Plot/Table Indicators

There are three different indicator variables in the plot data set to serve the following purposes:

- SUBIND: determines the number of indentations of the subtitle
- BOLDIND: determines if the subtitle is bold or normal weight
- SHADEIND: determines if the row in the plot has a shade background when *SHADING=1* or 2

These variables are automatically calculated by the macro.

Grouping Variables

There are four different grouping categories that the macro uses: *BY* variables, a *COLBY* (column by) variable, a *ROWBY* (row by) variable, and a *GROUPBY* (group by) variable. There can be multiple BY variables, but only one of each column by or row by variable. These are optional to increase the efficiency of doing subgroup analyses.

Variables for the *BY* groups include:

- BY_NUM: designates which *BY* variable the current data row represents. This is a numeric variable where the nth value represents the nth variable listed in the *BY* parameter.
- BY_LVL: designates the level of the current *BY* variable. This is a numeric variable where the nth value corresponds to the nth level of the variable. For example, if gender has the levels of 1=Male and 2=Female and BY_LVL=1 then would represent Male (the 1st level). A value of BY_LVL=0 indicates that the current row has the label for the *BY* variable in the SUBTITLE column.

There are not any variables added for the *COLBY* variable. Instead all of the summary and plot columns are duplicated for each level of the *COLBY* variable and a suffix (example: TOTAL_1) is added where the number represents which level of the *COLBY* variable.

Variables for the *ROWBY* groups include:

ROWBY_LVL: designates which level of the current ROW*BY* variable. This is a numeric variable where the nth value corresponds to the nth level of the variable. For example, if gender has the levels of 1=Male and 2=Female and ROWBY_LVL=1 then would represent Male (the 1st level).

Variables for the *GROUPBY* groups include:

- GROUPBY_LVL: designates which level of the current *GROUPBY* variable. This is a numeric variable where the nth value corresponds to the nth level of the variable. For example, if gender has the levels of 1=Male and 2=Female and GROUPBY_LVL=1 then would represent Male (the 1st level).

The GROUPBY_LVL variable is used for the discrete attribute map to color the scatter plot and highlow plot if *GROUPBY* is specified.

CREATING THE FOREST PLOT

The macro uses the Graph Template Language within the TEMPLATE procedure in combination with the SGRENDER procedure to produce the final graph. The template splits the graph into sections or panels: the subtitle panel, the plot panels, and the statistical summary panels using the LATTICE layout within GTL. The lattice creates a plot space that has one column for each item being displayed by the DISPLAY parameter and one row for each value of the *ROWBY* variable. Each column is considered a section or panel.

Due to the evolution of the Graph Template Language from SAS 9.2 and 9.4+, producing a forest plot has become much easier with new functionality such as AXISTABLE and TEXTPLOT. However the MVMODELS does not use these new features in favor of primarily using annotation.

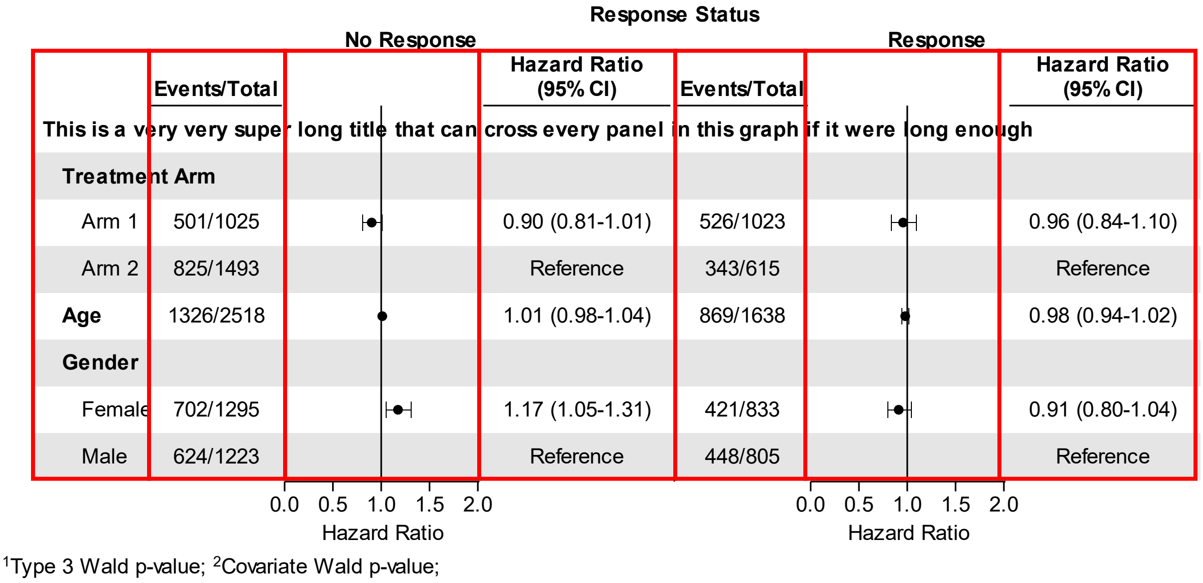
Annotation versus Data Driven Plotting

SAS 9.3 introduced annotation to the SG procedures and with it the DRAW functions within GTL. The draw functions, DRAWTEXT, DRAWLINE, DRAWARROW, and DRAWPOLYGON, allow the user to manually annotate a graph with text, lines, and other shapes. The creation of these annotations is generally more tedious and much less flexible compared to data driven graphing, but they have a distinct advantage within multi-panel graphs which is why the MVMODELS macro uses the annotation method instead of being entirely data driven.

Annotation can cross over panels

One of the greatest challenges of designing a forest plot is allocating enough space to each section so that the graph is easy to read and none of the text involved gets cut-off. Annotation text does not get cut-off by the graph space and allows the text to flow across multiple panels. This comes in handy for having long subsection headers, model titles, or longer estimate values.

Figure 12. Displays an example of annotation crossing over panels versus the graph getting cut-off.



Response Status

	No Response			Response		
	Events/Total		Hazard Ratio (95% CI)	Events/Total		Hazard Ratio (95% CI)
This is a very very super long title that can cross every panel in this graph if it were long enough						
Treatment Arm						
Arm 1	501/1025		0.90 (0.81-1.01)	526/1023		0.96 (0.84-1.10)
Arm 2	825/1493		Reference	343/615		Reference
Age	1326/2518		1.01 (0.98-1.04)	869/1638		0.98 (0.94-1.02)
Gender						
Female	702/1295		1.17 (1.05-1.31)	421/833		0.91 (0.80-1.04)
Male	624/1223		Reference	448/805		Reference

[1]Type 3 Wald p-value; [2]Covariate Wald p-value;

Figure 12. The red panels indicate where the layout panels end for each column. The model title is able to stretch across all seven panels in this example, where a normal data driven graph it would get cut off within the first panel instead.

The macro facility takes away the tediousness of annotation

Utilizing the macro facility removes the tediousness of writing out each annotation separately and fully automates the process. The MVMODELS macro pulls the values from the plot data set along with the coordinates to plot them and writes each value within the GTL environment without input from the user.

Annotation allows the use of Unicode, superscript and subscript

The DRAWTEXT function allows the use of Unicode characters, superscripts and subscripts in the text it creates. Normal data-driven graph elements such as labels and AXISTABLE values do not handle superscripts and subscripts.

Limitations of annotation

The greatest limitation that annotation offers is that the graph does not allocate space automatically to fit annotation. This means that the programmer must use other means to define graph space in order to properly show the annotation. This can be as simple as printing a blank space of the same size and font of the DRAWTEXT function in the spot the user wants the text.

Graph Section/Panel Descriptions

Subtitle Panel

The subtitle panel contains the row headers for the other two panels and includes: the model titles, the covariate labels, and the levels of the covariate. The macro creates the forest plot in a row-by-row basis. Each row of the graph will have a row header (or subtitle), at least one possible scatterplot, and optional columns of summary statistics. Multiple models can be run and displayed in the plot, and model titles go before any covariates from a model are listed. Covariate labels are listed before the covariate levels. For example, for gender there would be a row with a label such as "Gender" followed by up to two rows (depending on display options) designating a "Male" row and a "Female" row.

The subtitles are drawn entirely with DRAWTEXT statements, which like ENTRY statements can be left aligned and can have spaces added in front for indentation. Unlike ENTRY statements however, the coordinates for the text can be specified with the DRAWTEXT statement. This allows the text to be aligned with the y-axis from the plot panel. Each subtitle has its own DRAWTEXT statement. DRAWTEXT statements are not bound by the walls of a layout or lattice cell, so longer model titles or variable labels can fit with less white space in the graph.

Plot Panels

There are only three components to the plot: the scatterplot, the confidence bounds, and the reference line. The reference line is drawn with a REFERENCELINE statement. The scatterplots are drawn with the SCATTERPLOT statements, and the confidence bounds are drawn with the HIGHLOWPLOT statement. The HIGHLOWPLOT statement is used instead of the error bar options within the SCATTERPLOT statement for two reasons. The first is that when the line thickness is increased the endcaps on the error bars with SCATTERPLOT increase dramatically in vertical height which looks unprofessional. The error bar endcaps stay the same size or increase at a much slower rate when the line thickness is increased with the HIGHLOWPLOT statement. The second reason is that there is an option in the HIGHLOWPLOT statement to point to variables that determine if the error caps are drawn. The option to point to a variable that determines if an error bar endcap is drawn is helpful as the Data step can be used to determine if the bars exceed the *MIN* or *MAX* parameters and programmed accordingly with IF/ELSE logic. The macro does not draw the error bar endcaps if the error bars extend beyond the maximum or minimum values of the axis.

Each model created by the macro can be colored separately, sized differently, and have unique symbols. This is done by building a discrete attribute map and applying it to the model number variable.

Statistical Summary Panel

The statistical summary panels are drawn similarly to the subtitle panel, but without the need for indenting or changing font weights. They are centered within their column of the LATTICE layout and drawn with the DRAWTEXT statement.

Reference guides

Reference guides are text that describes what an area of the graph means compared to a reference line (e.g. “Males perform better” or “High-grade perform worse”). These are generally paired up with an arrow to show the direction that the reference guides are referring to. The MVMODELS macro creates these with a combination of DRAWTEXT and DRAWARROW. Additional rows are added into the graph in order to allocate space. The text that is printed is directly input by the user with the *REFGUIDELOWER* and *REFGUIDEUPPER* parameters. The following is an example of adding reference guides:

```
%mvmodels(DATA=random, METHOD=survival, TIME=os_time, CENS=os_stat, CEN_VL=0,
    COVARIATES=arm, by=gender, TYPE=2, CAT_DISPLAY=5, pval_type3=0,
    HEIGHT=4in, REFLINE=1,
    REFGUIDELOWER=Favors Arm 1,REFGUIDEUPPER=Favors Arm 2);
```

Figure 13. Example of adding reference guides to bottom of the plot space

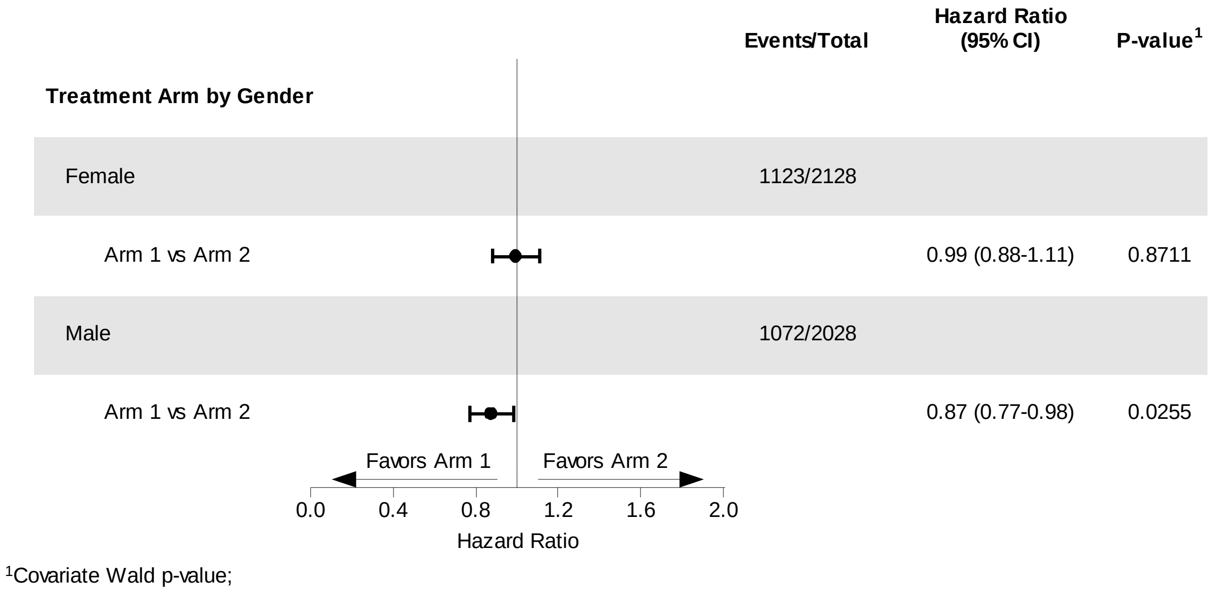


Figure 13. The reference guides originate from the reference line and point towards the minimum and maximum values. The text is manually specified by the user.

The reference guides can also be printed at the top of the graph space, and line breaks are inserted with the ` symbol.

```
%mvmodels(…,REFGUIDEVALIGN=top,REFGUIDEHALIGN=in,REFGUIDELINEARROW=open,
    REFGUIDELOWER=Favors`Arm 1,REFGUIDEUPPER=Favors`Arm 2);
```

Figure 14. Example of adding reference guides to top of the plot space

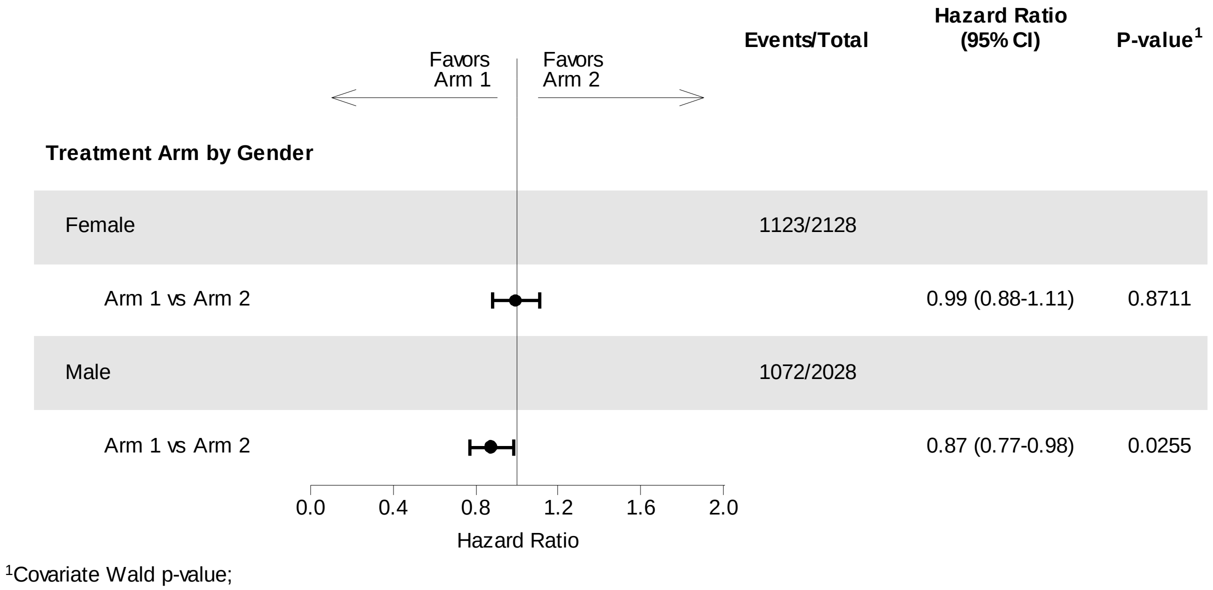


Figure 14. REFGUIDEVALIGN=top moves the reference guides to the top. REFGUIDEHALIGN=in aligns the reference guides against the reference line. The ` symbol creates line breaks.

CREATING THE TABLE

The table is printed using the REPORT procedure due to the customizability the procedure offers. The capability to make compute variables on-the-fly, make style modifications on-the-fly, and create spanning headers easily makes the REPORT procedure the go-to procedure for creating high quality analysis output.

Designing a style template

The MVMODELS macro is able to output to several ODS destinations including LISTING, RTF, PDF, EXCEL and POWERPOINT. Each of these destinations has its own programming quirks and challenges, and so the macro generalizes wherever possible but ultimately has separate code adjustments for each destination. The macro creates a style template using the TEMPLATE procedure to set up a majority of the output table, and then fine-tunes the final REPORT procedure depending on the destination. One REPORT procedure is ran for each destination currently open.

Adding spaces between columns

The REPORT procedure allows the creation of COMPUTE variables which are not actually in the input data set. These do not actually have to contain a value however, and so the macro uses COMPUTE variables as “dummy” variables. They are assigned a missing value, and given a certain width to create a space between different values of the *COLBY* variable.

Creating spanning headers

The REPORT procedure allows one variable to be used multiple times and renamed. The macro uses a blank variable as an across variable multiple times in order to add a header across selected variables. These can be nested in order to create multiple spanning headers.

Styling the SUBTITLE column

The MVMODELS macro creates an indicator variable for whether the value should be bold or regular font weight, and creates a variable for how many indents the text should have. A COMPUTE block within the REPORT procedure is used to add these style modifications to the SUBTITLE column.

Adding shading to rows

The MVMODELS macro creates a variable indicator to whether a row should be shaded or not. A COMPUTE block within the REPORT procedure allows a style modification to be done to an entire row depending on this indicator variable.

Creating the table in the LISTING destination

The LISTING destination is a much tougher destination to make easy to read tables as length of variables is very important. Typically variables are made with lengths/formats that are longer than necessary to contain all of the information. Within the LISTING destination however this just adds more blank space to a column. The MVMODELS macro creates a separate dataset for printing to the LISTING destination where it finds the longest value in any character variable and sets the length of the variable to the length of this value.

In order to create lines that cross the whole page the repeat function is used along with the current LINESIZE option value. The REPEAT function allows for a character to be repeated n number of times, so by combining the hyphen with the LINESIZE value a dashed line will cross the entire page. A similar technique is used to create the underlines in each column header. A dashed line the length of each variable plus four (for space between columns) is drawn under each column label. Spanning headers are added to the COLUMNS statement with a line drawn underneath that is the length of the sum of the variables in the spanning header.

CONCLUSION

The MVMODELS macro is a powerful and flexible tool that was built to handle the large variety of modeling prevalent to clinical oncology trials. The macro outputs both a professional looking graph and summary table to a multitude of ODS destinations and has many options to fine-tune the appearance to match the user's preference. The method used to create the forest plot is applicable to many other types of graphs. MVMODELS is an incredibly useful macro for any programmer performing survival or logistic regression analyses.

REFERENCES

[1] Therneau T (2014). _A Package for Survival Analysis in S_. R package version 2.37-7, <URL: http://CRAN.R-project.org/package=survival>.

[2] cNeil BJ: The meaning and use of the area under a receiver operating characteristic (ROC) curve. Radiology 143:29-36, 1982.

CONTACT INFORMATION

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

Name: Jeffrey Meyers
Enterprise: Mayo Clinic
Address: 200 First Street SW
City, State ZIP: Rochester, MN 55905
Work Phone: 507-266-2711
E-mail: Meyers.jeffrey@mayo.edu / jpmeyers.spa@gmail.com
Website (Macro available for download): SAS Communities Page