PharmaSUG 2019 - Paper BP-84

Useful SAS techniques in Efficacy Analysis for Oncology studies

Joy Zeng, Pfizer Inc;

ABSTRACT

Oncology refers to the research on prevention, diagnosis and treatment of cancer. Oncology studies are, in general, more complex than studies in other therapeutic fields. This paper summarizes the primary sources of complexity, including endpoints, data collection, AE reporting, tumor assessment under RECIST, oncology-specific domains, and special statistical analysis for efficacy data. This paper also discusses multiple oncology-related statistical methods (e.g. Cox regression, Kaplan-Meier, log-rank tests) and graphical data representations (e.g. waterfall plots, bar charts, mean standard error plots, spaghetti plots, and forest plots). Finally, the relevant *SAS*® code is given for all of these methods and representations, with the goal of providing statistical programmers the necessary knowledge and tools for creating and validating tables and figures from oncology studies.

INTRODUCTION

Analyzing oncology studies is challenging due to the complex design and endpoints of studies within oncology. Statistical programmers generate oncology-related analysis data structures, produce meaningful tables, figures, and listings containing oncology-specific data based on statistical analysis plan (SAP) and specifications. However, this can be difficult for programmers who are unfamiliar with oncology-specific practices and guidelines. This paper gives an introduction to oncology-specific studies, endpoints, and its unique statistical analysis techniques, in order to provide statistical programmers with the necessary background knowledge to work in oncology studies.

INTRODUCTION TO ONCOLOGY STUDIES

Studies in oncology and those in other therapeutic fields differ in many ways. The following list summarizes some of the key differences that statistical programmers that are new to the field should understand:

1. **Endpoints**: In oncology studies, the most commonly used endpoints are objective response rate (the proportion of responders, complete or partial, among all eligible subjects), overall survival (OS, time from randomization to death from any cause), progression-free survival (PFS, time from randomization to disease progression or death), and quality of life (QOL) [1].
2. **Data Collection:** In addition to standard safety data, oncology trials require more information to be collected in the CRFs to evaluate the efficiency of the trials. This information includes tumor measurements, their responses, and ecog performance statuses.
3. **Adverse Event Reporting:** The National Cancer Institute (NCI) has developed oncology-specific guidelines for adverse event reporting, known as the National Terminology Criteria for Adverse Events (CTCAE). In CTCAE, each AE is rated on a scale from 1 to 5, depending on severity (grade 1 corresponds to mild and grade 5 corresponds to death). Every AE is also coded to preferred term and system organ class using the Medical Dictionary for Regulatory Activities (MedDRA) and additionally, classified by severity scales via CTCAE.

   For safety analysis, treatment-emergent adverse events (TEAEs) are commonly the main focus. The treatment-emergent period is typically defined as the period of time from the first dose date on a study drug to a pre-specified period of time (e.g. 28 days, 30 days) after the last dose date. In a crossover study (cross over from placebo to study drug), the derivation of treatment-emergent period can be more complex since it's needed to be carefully determined which treatment is the truly trigger for the AEs. The incidence rates of TEAEs are typically summarized by system organ class (SOC) and preferred term (PT) in terms of severity, relationship to the drug, cause of dose reduction, etc.
4. **Tumor measurement and assessment under RECIST guidelines:** In oncology, RECIST is the primary tool used to access tumor progression or shrinkage for solid tumor.

In RECIST [2], tumor lesions are first categorized as measurable or non-measurable at baseline. Among the measurable lesions, target lesions are identified and recorded the baseline measurements.

Figure 1 is a demonstration for categorizing tumor lesions.

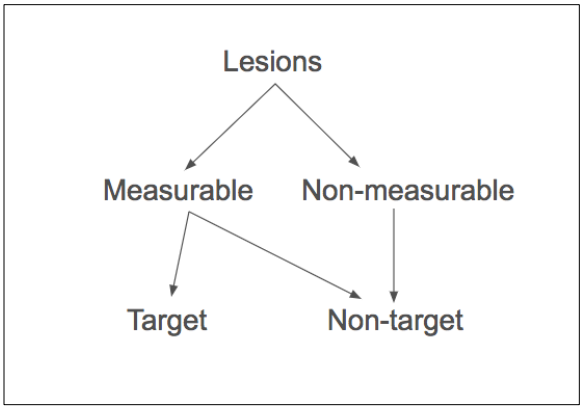

Figure 1 Measurable vs. Non-measurable Lesions

During treatment, subsequent measurements are performed for all target, non-target, and new lesions at each pre-specified time-point. The changes in tumor size determine tumor response, and the response at each time point is evaluated as follows:

For target lesions:

- Complete Response (CR): Disappearance of all target lesions.
- Partial Response (PR): At least a 30% decrease in the sum of diameters of target lesions, taking as reference the baseline sum diameters.
- Progressive Disease (PD): At least a 20% increase in the sum of diameters of target lesions, taking as reference the smallest sum on study; or the appearance of one or more new lesions.
- Stable Disease (SD): Neither sufficient shrinkage to qualify for PR nor sufficient increase to qualify for PD.

For non-target lesions:

- Complete Response (CR): Disappearance of all non-target lesions and normalization of tumor marker level.
- Non-CR/Non-PD: Persistence of one or more non-target lesion(s) and/or maintenance of tumor marker level above the normal limits.
- Progressive Disease (PD): Unequivocal progression of existing non-target lesions or appearance of one or more new lesions.

Per RECIST, the best overall response at the subject level is the best response recorded from the start of the treatment until disease progression, taking confirmation requirements into account.

5. **Oncology specific SDTM domains:** TU(Tumor Identification), TR (Tumor Results), RS (Response), and Time-to-event ADaM datasets
6. **Censoring and confirmation rules:** Censoring occurs when patients haven't experience any event of interest, or have not had a follow-up. Most time-to-event endpoints require carefully reviewing the censoring rules and criteria for confirmation of progression.

For example, one of the most commonly used censoring rules for OS and PFS can be summarized as follows:

Table 1 is a summary of censoring rules.

Endpoint	Date of Event	Date of Censoring
OS	death date	date of last known alive
PFS	date of progression	date of last assessment/scan

Table 1 Censoring Rules for OS and PFS as endpoints

In time-to-event ADaM dataset, the information of event or censoring is captured by the variable CNSR, with CNSR = 0 for events and CNSR > 0 for censored records. Additionally, date of event/censoring is collected in ADT variable, associated with AVAL variable computing the time to event/censoring from the origin. The general formula to calculate AVAL is typically (ADT [Date of Event/Censoring] - randomization date + 1).

Finally, confirmation criteria and pre-defined time windows for confirmatory scans need to be documented.

Figure 2 is a demonstration for deriving CNSR and AVAL variables.

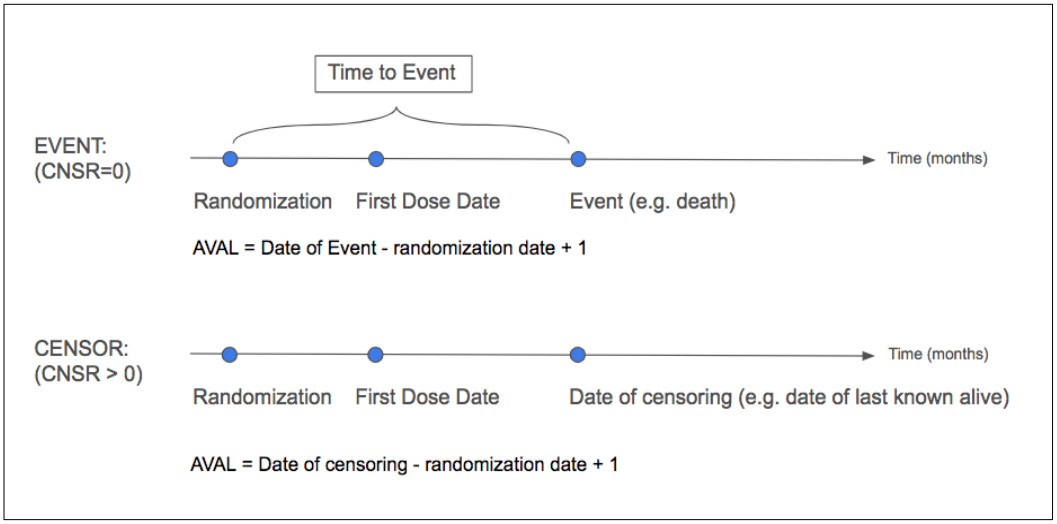

Figure 2 Derivation of CNSR and AVAL

Table 2 is an example of ADTTE dataset, where OS is the event of interest.

SUBJID	PARAM	PARAMCD	RANDDT	ADT	AVAL	CNSR	EVNTDESC	EVNDTDSC
101	Time to Overall Survival	OS	2018-02-06	2018-12-02	9.85	0	Dead	Date of Death
102	Time to Overall Survival	OS	2018-02-11	2018-08-26	6.47	1	Alive	Date of Last Known Alive
103	Time to Overall Survival	OS	2018-02-13	2018-09-23	7.32	1	Lost to Follow-up	Date of Last Known Alive
104	Time to Overall Survival	OS	2018-02-22	2018-12-21	9.95	1	Dead after Analysis Cutoff	Analysis Cutoff Date

Table 2 An Example of ADTTE Dataset

Note that AVAL is calculated by analysis date in month: (ADT-RANDDT+1)/30.4375

7. **Event-oriented efficacy analysis:** efficacy analyses in oncology trials are typically event triggered and will be evaluated only at the times when occurrence of pre-specified event counts [3]. For instance, an interim analysis and a final analysis for OS endpoint occurs when a pre-specified number of death events in protocol has been reached.
8. **Special statistical analysis for efficacy:** Oncology trials involve special statistical techniques to analyze their efficacy. Some of these techniques include:
    - log-rank test to compare the OS/PFS between two treatment groups
    - Kaplan-Meier curve is an intuitive graphical representation of the survival distribution in different treatment groups. In addition, 50th percentile of KM estimates can be used as the estimate of the median duration (time when half of the patients are event free) for time-to-event endpoints
    - Reserve Kaplan-Meier, which refers reversing events such that loss-to-follow-up are treated as "events" while the outcome events are treated as "censored", to estimate median follow-up time
    - Cox regression provides us a way to estimate and compare the survival experiences of two treatment groups. We typically calculate Hazard ratio associated with Cox regression, which refers to the relevant risk of experiencing an event of interest between two groups.
    - Binomial test of binary proportion along with its confidence interval (Exact CI or normal approximation) to compare the proportion of subjects free of event or the difference in response rates at given landmark time points

USEFUL SAS TECHNIQUES IN ONCOLOGY STUDIES

Efficacy time-to-event endpoints such as overall survival (OS), PFS (progression-free survival), and time to PSA (Prostate-specific Antigen) progression are typically evaluated by survival analysis. Survival analysis takes into account the censoring observations (e.g. lost to follow up or no event occurs) and the survival time refers to the time from starting point (e.g. randomization) to the occurrence of an event of interest (e.g. death). There are three major SAS procedures for survival analysis [4]:

PROC LIFETEST - produces life tables and Kaplan-Meier estimates. KM estimate is a non-parametric estimate of the survival function and no assumption needed in either underlying hazard function or proportional hazard

PROC PHREG - Cox regression makes assumptions on proportional hazard and models the effect of predictors and covariates. It handles both continuous-time (EXACT method) and discrete-time (DISCRETE method) data and allows for time-dependent covariables

PROC LIFEREG - parametric regressions which make assumptions on the distribution of time-to-event variables (such as Weibull, exponential, or lognormal) and model the underlying hazard/survival function

OBJECTIVE RESPONSE RATES ANALYSES

Objective response rate is defined as the proportion of responders (Per RECIST, a response refers to a subject's having CR or PR as their best overall response). Typically, objective response rates and associated 95% confidence intervals (CI) are calculated for each arm. The difference in the response rates between two treatments, and the associated 95% CI and p-value (e.g. based on the Cochran-Mantel-Haenszel Test) are calculated as well. An example mock-up table to summarize objective response rates is shown below:

Figure 3 is an example mock-up table to summarize objective response rates

	Drug A	Placebo	Treatment Comparison
Best Overall Response			
Complete Response (CR)	XX (XX%)	XX (XX%)	
Partial Response (PR)	XX (XX%)	XX (XX%)	
Stable Disease (SD)	XX (XX%)	XX (XX%)	
Progressive Disease (PD)	XX (XX%)	XX (XX%)	
Best Objective Response (CR or PR)	XX (XX%)	XX (XX%)	
95% CI for Objective Response Rate	XX% - XX%	XX% - XX%	
Difference in Objective Response Rate			XX%
95% CI for Difference in Objective Response Rate			(XX% - XX%)
P-value			XXX

Figure 3 Mock-up Table for Summarizing Objective Response Rates

```
**best objective response is defined as CR or PR**
CR - 1
PR - 2
SD - 3
PD - 4
NE - 5;
data adrs;
   set adrs;
   if aval in (1,2) then objrespfl = 0;
   else if aval ne . then objrespfl = 1;
run;
** Best overall response **;
proc freq data = adrs;
   by trt01pn;
   tables objrespfl/out = rs0(drop=percent);
run;
** 95% CI for ORR based on binomial test (Clopper-Pearson CI)**;
proc freq data = adrs;
   by trt01pn;
   table objrespfl/binomial(exact);
   ods output binomialcls = rs1;
run;
** Difference in response **;
proc freq data = adrs;
   tables trt01pn*objrespfl/riskdiff;
*The option riskdiff computes the difference based on standard normal
approximation;
   ods output RiskDiffCol1 = rs2;
run;
** p-value based on CMH test**;
proc freq data = adrs;
   tables trt01pn*objrespfl/cmh;
   ods output cmh=cmh (where=(AltHypothesis='Row Mean Scores Differ'));
run;
```

Table 3 is a result metadata corresponding to Figure 3

Metadata Field	Metadata
DISPLAY IDENTIFIER	Table 1.1
DISPLAY NAME	Best Overall Soft Tissue Response
PARAM	Soft Tissue Best Overall Response
PARAMCD	STOVRESP
ANALYSIS VARIABLE	AVAL
ANALYSIS PURPOSE	Co-Primary Analysis
DATASET	ADRS
SELECTION CRITERIA	ITTFL="Y" and PARAMCD="STOVRESP"
DOCUMENTATION	SAP Section X.XX

PROGRAMMING STATEMENTS	(NOTE: objrespfl=1 FOR AVAL=3, 4 OR 5 AND objrespfl=0 if AVAL=1 OR 2)
	proc freq data = adrs;
	by trt01pn;
	tables objrespfl/out = rs0(drop=percent);
	run;
	proc freq data = adrs;
	by trt01pn;
	table objrespfl/binomial(exact);
	ods output binomialcls = rs1;
	run;
	proc freq data = adrs;
	tables trt01pn*objrespfl/riskdiff;
	ods output RiskDiffCol1 = rs2;
	run;
	proc freq data = adrs;
	tables trt01pn*objrespfl/cmh;
	ods output cmh=cmh (where=(AltHypothesis='Row Mean Scores Differ'));
	run;

Table 3 Result Metadata Corresponds to Figure 3

ANALYSES ON TIME TO EVENT

Examples of time to event analyses are duration of overall survival, duration of progression-free survival, and time to progression. The 50 percentile of KM estimates and the associated confidence intervals are typically used for estimating the median duration of those time-to-events. Two treatment arms are compared using stratified or unstratified log-rank test, and if the p-value is less than pre-specified type I error rate (e.g. 0.05), it indicates statistical significance and concludes the drug prolongs time to event (e.g. overall survival or time to progression). The hazard ratio is computed based on a Cox regression model (which can be stratified or unstratified, and use different covariates in the model). A hazard ratio (relative to a placebo) with value less than 1 indicates favoring the experimental treatment arm. Notice that in the code sample below, setting trt01pn = 1 for treatment and trt01pn = 2 for placebo, yields a different hazard ratio than setting the variables the other way around, because of the difference in reference groups. Therefore, carefully reviewing the footnotes in the mock-ups about the calculation of the hazard ratio is essential in properly analyzing the data.

Figure 4 is a sample mock-up for duration of OS, duration of PFS, and time to progression.

	Drug A	Placebo	Treatment Comparison
Time to event			
Median (95%CI)	XX (XX,XX)	XX (XX,XX)	
P-value			XXX
Hazard Ratio (95%CI)			XX (XX,XX)

Figure 4 Mock-up Table for Time-to-Event

```
** Median survival time and p-value **;
ods listing close;
ods output Quartiles=_quart(keep=trt01pn estimate lowerlimit upperlimit
percent
                          where=(percent eq 50))
        HomTests=pval(keep=test probchisq where=(test='Log-Rank'));
   proc lifetest data = adttee;
   time aval*cnsr(1);
   strata trt01pn;
   run;
ods output close;
ods listing;

** Hazard ratio **;
ods listing close;
ods output parameterestimates=hzdata(keep=hazardratio hrlowercl
hruppercl);
   proc phreg data = adttee;
   class trt01pn;
   model aval*cnsr(1)=trt01pn/ties=discrete risklimit;
   hazardratio trt01pn;
   run;
ods output close;
```

Time to disease progression is sometimes evaluated by **Nadir**, defined as the lowest value or best response prior to current assessment. For instance, for the analysis of time to PSA progression, Nadir is defined as the lowest PSA value prior to current measurement and is derived separately for each time point. Based on the calculated Nadir value, PSA progression can be defined as, for example, PSA value with an increase above NADIR >=30% which is confirmed with a consecutive value that is also increase above NADIR>=30%. Accordingly, ADT for events can be defined as the date of first confirmed PSA progression. When developing SAS programs, using RETAIN statement and lag function can be simple and efficient to compare values across multiple records. The following code is an example of deriving Nadir and time-to-event analysis dataset.

```
data psa;
format RANDDT mmddyy10.;
input USUBJID $ 1-3 AVISIT $ 5-12 AVISITN LBDTC $ 17-26 PSA;
retain BASE;
RANDDT='15SEP2018'd;
if avisitn=0 then base=psa;
if avisitn > 0 then postbfl=1; else postbfl=0;
datalines;
101 BASELINE 0  2018-09-21 23.31
101 WEEK 1   1 2018-09-27  44.65
```

```
101 WEEK 4   2 2018-10-09  21.78
101 WEEK 9   3 2018-11-13  13.96
101 WEEK 15  4 2018-12-27  31.34
101 WEEK 19  5 2019-01-23  26.51
;
run;
proc sort data=psa;
by usubjid postbfl;
run;

** Deriving NADIR **;
data psa;
set psa;
by usubjid postbfl;
retain NADIR;
if first.usubjid then nadir=.;
lagpsa=lag(psa);
if postbfl then do;
  if first.postbfl then nadir=base;
  else nadir=min(nadir,lagpsa);
end;
if postbfl and psa ne . then do;
  chg_nadir=psa-nadir;
  if nadir ne 0 then pchg_nadir=chg_nadir/nadir*100;
end;
run;

** Deriving variables ge30pct_nadir and nextge30pct_nadir **;
data psa;
set psa;
retain nextge30pct_nadir;
by usubjid avisitn;
ge30pct_nadir=(pchg_nadir>=30);
nextge30pct_nadir=lag(ge30pct_nadir);
run;

** Deriving time-to-event analysis dataset **;
data adpsa;
format adt date9.;
set psa;
if ge30pct_nadir and nextge30pct_nadir;
ADT=input(lbdtc,yymmdd10.);
AVAL=(ADT-RANDDT+1)/30.4375;
EVNTDESC='PSA Progression';
CNSR = 0;
keep usubjid lbdtc adt randdt aval evntdesc cnsr;
run;
```

Figure 5 is an example dataset demonstrating how NADIR is derived

Obs	USUBJID	RANDDT	AVISIT	AVISITN	LBDTC	PSA	BASE	postbfl	NADIR	lagpsa	chg_nadir	pchg_nadir	ge30pct_nadir	nextge30pct_nadir
1	101	09/15/2018	BASELINE	0	2018-09-21	23.31	23.31	0	.	.	.	.	0	.
2	101	09/15/2018	WEEK 1	1	018-09-27	44.65	23.31	1	23.31	23.31	21.34	91.549	1	0
3	101	09/15/2018	WEEK 4	2	018-10-09	21.78	23.31	1	23.31	44.65	-1.53	-6.564	0	1
4	101	09/15/2018	WEEK 9	3	018-11-13	13.96	23.31	1	21.78	21.78	-7.82	-35.904	0	0
5	101	09/15/2018	WEEK 15	4	018-12-27	31.34	23.31	1	13.96	13.96	17.38	124.499	1	0
6	101	09/15/2018	WEEK 19	5	019-01-23	26.51	23.31	1	13.96	31.34	12.55	89.900	1	1

Figure 5 An Example Dataset for Demonstrating the Derivation of NADIR

Figure 6 is a time-to-event analysis dataset based on the example dataset in Figure 5

Obs	USUBJID	LBDTC	ADT	RANDDT	AVAL	EVNTDESC	CNSR
1	101	2019-01-23	23JAN2019	09/15/2018	4.30390	PSA Progression	0

Figure 6 Analysis Dataset for Time to PSA Progression

ANALYSES ON FOLLOW-UP TIME

To estimate the follow-up time between two treatment arms, the median follow-up time can be estimated by Reverse Kaplan-Meier as mentioned above. An example summary table to present summary statistics is shown below:

Figure 7 is a sample mock-up for follow-up time.

	Drug A	Placebo
Follow-up Time Based on Reverse Kaplan-Meier Estimates		
n	XXX	XXX
25th Percentile	XXX	XXX
Median	XXX	XXX
75th Percentile	XXX	XXX

Figure 7 Mock-up Table for Follow-up Time

```
/*To get total number*/
proc means data = adeff nway noprint;
   var aval;
   class trt01pn;
   output out = tot n=n;
run;
proc sort data = tot;
   by trt01pn;
run;
proc transpose data = tot
           out = tran_tot (drop=_label_
                           rename=(_name_=name _1=drug_A _2=placebo));
   id trt01pn;
   var n;
run;
/*To get the percentile*/
%macro reverse_km(num=, dsout=)
ods listing close;
ods output Lifetest.Stratum&num..TimeSummary.Quartiles = &dsout.;
   proc lifetest data = adeff;
        time aval*cnsr(0);
         strata trt01pn;
   run;
ods output close;
ods listing;
proc sort data = &dsout.;
   by percent;
run;
%mend;
%reverse_km(num=1, dsout= trt1);
%reverse_km(num=2, dsout= trt2);
```

Reversing “events” and “censored” to estimate median follow-up time

```
data percent;
    merge trt1(keep=percent estimate rename=(estimate = drug_A))
                trt2(keep=percent estimate rename=(estimate = placebo));
     by percent;
     rename percent=col1;
run;
```

GRAPHICAL ANALYSIS

This section contains a list of commonly used graphical analysis techniques in SAS.

Kaplan-Meier Curve

KM curves are widely used to compare the survival experience of two treatment arms and provide an intuitive graphical presentation of the proportion of patients who remain event-free at each time point. A summary table of patient at-risk, which refers to those who have not yet experienced an event of interest (e.g. disease progression) or censored is often presented along with the KM curves. The SAS command "ATRISK" adds the number of individuals still at risk.

Figure 8 is an example of KM Curve

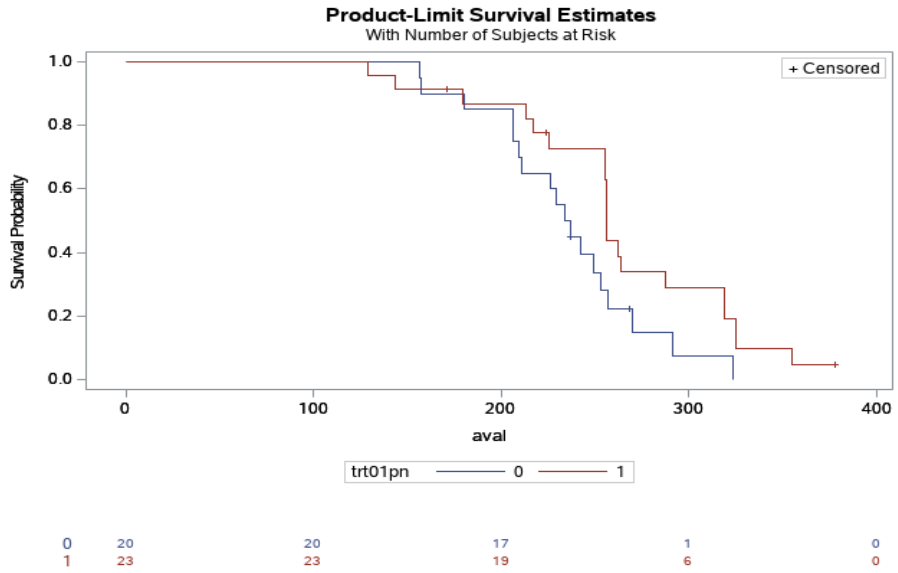

Figure 8 Example of KM Curve

```
ods graphics on;
ods output ProductlimitEstimates = _ple
                Quartiles = _quart
                CensoredSummary = _cs
                Survivalplot = _splot
                homtests = pval;
                **_ple: Event/Cum. Event
                  _quart: estimate of the median survival time
                  _cs: censored summary
                  _splot: info about patient at risk
                  pval: p-value based on log rank test;
   proc lifetest data=adttee method=km plots=(survival(atrisk
   (outside(0.15))) LS);
```

```
        time aval*cnsr(1);
        strata trt01pn;
    run;
ods output close;
ods graphics off;
```

Note that it is important to carefully review the shape of KM curves and the labeling in the legend when validating KM curves. For example, if we have a hazard ratio relative to placebo that is less than 1, then the KM curve of the treatment arm would expect to be above the KM curve of the placebo arm. This is a critical aspect to keep in mind when validating the figures.

Exam the Assumption of Proportional Hazards

The Cox regression model is based on the assumption of proportional hazards such that the hazard for any individual is a fixed proportion of the hazard for any other individual. Graphically, such assumption is supported by parallel lines in the graph of the -log(S(t)) versus t OR log(-log(S(t))) versus log(t). To output the graph in SAS, we can specify PLOTS=(LS, LLS) in the PROC LIFETEST statement, as seen in the example code above.

Figure 9 is a plot of -log(S(t)) versus t.

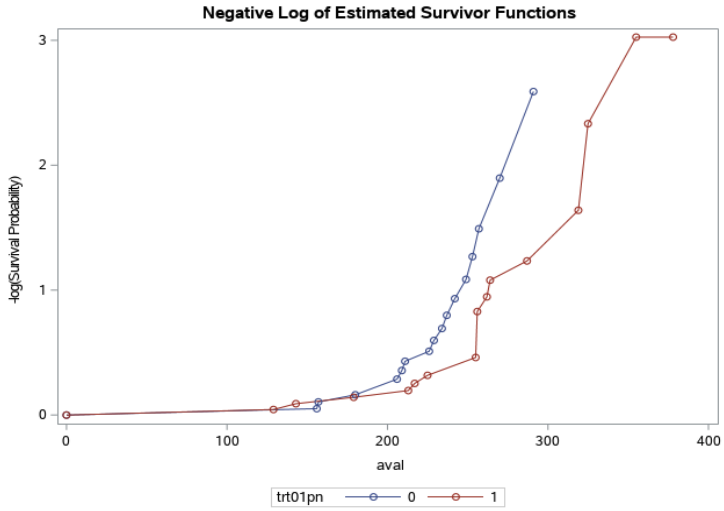

Figure 9 Plot of -log(S(t)) versus t

In Figure 7, none of the curves is a straight line, which indicates the hazards are not constant, but rather, increasing with time.

Waterfall Plot

Waterfall plots are a commonly used graphical presentation in oncology. Waterfall graphs help programmers visualize the change in parameter of interest (e.g. tumor size, PSA change from baseline) for each individual subject. Each vertical bar represents one subject in the study. Scanning the x-axis from left to the right shows the worst values to the best values (e.g. percent increase in tumor size to percent reduction in tumor size). From this perspective, the analysis dataset used in creating a waterfall plot should be one subject per record and ordered based on the parameter of interest. For instance, in Figure 8, the best percentage change from baseline in PSA (maximum percentage reduction in PSA from baseline) is graphically summarized in waterfall plot.

Figure 10 is an example of a waterfall plot.

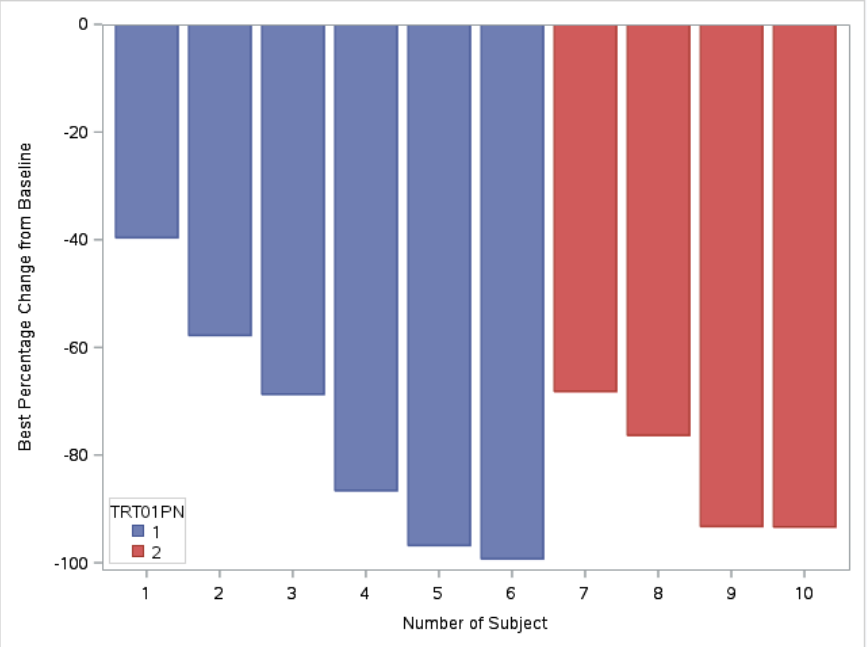

Figure 10 Example of a Waterfall Plot

```
data adpsa;
format SUBJID z4. ;
input  SUBJID PCHG TRT01PN;
datalines;
   1001 -86.56  1
   1001 -86.56  1
   1001 -83.21  1
   1002 -95.23  1
   1002 -93.28  1
   1002 -94.59  1
   1002 -96.74  1
   1003 -10.34  1
   1003 -39.57  1
   1004 -93.31  2
   1004 -91.73  2
   1005 -68.67  1
   1006 -57.72  1
   1006 -52.73  1
   1007 -99.23  1
   1007 -98.54  1
   1007 -95.35  1
   1008 -76.29  2
   1009 -68.14  2
   1010 -93.21  2
   1010 -92.57  2
;
run;
/*preparing dataset*/
proc sort data = adpsa;
by usubjid pchg;
```

```
where PCHG ne . ;
run;

data adpsa;
set adpsa;
by subjid pchg;
if first.usubjid;
proc sort; by trt01pn descending pchg;
run;

data bestpsa_;
set adpsa;
n = _n_;
run;
/*creating graph*/
ods listing close;
ods rtf style=basic file='&path/psa_waterfall.rtf'
style=styles.statistical;
ods graphics on;
proc sgplot data = bestpsa_;
vbar n/ response = pchg group = trt01pn;
xaxis label = 'Number of Subject' fitpolicy=thin;
yaxis label = 'Best Percentage Change from Baseline';
keylegend/location inside down=2;
run;
ods graphics off;
ods tagsets.rtf close;
ods listing;
```

Bar Chart

Bar charts are a great way to display categorical data and summarize the frequency and percentage for each category. For example, we can summarize the best overall response in a bar chart (shown below) and add a summary table in the legend to display the statistics of the objective response rate.

Figure 11 is a bar chart summarizing the best overall response for Drug vs. Placebo

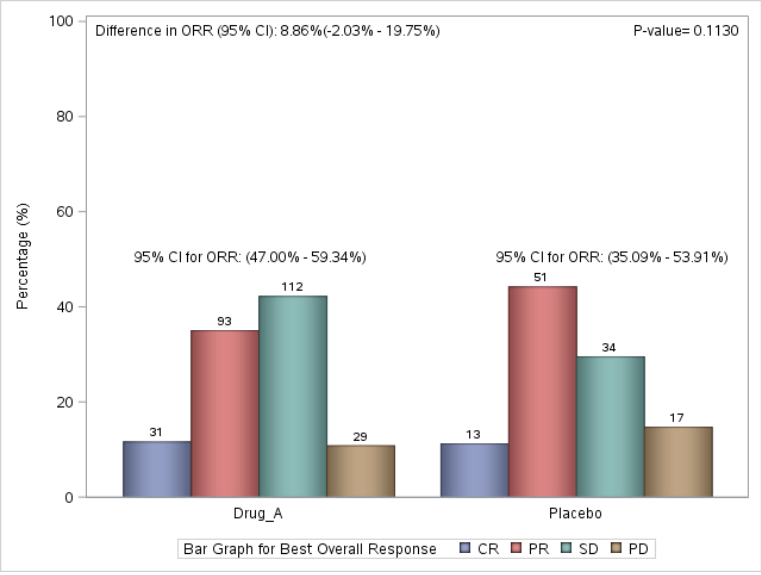

Figure 11 Bar Chart for Drug vs. Placebo

```
   data adpsa;
       input trt01p $10. aval $ count;
      datalines;
      Drug_A     CR 31
      Drug_A     PR 93
      Drug_A     SD 112
      Drug_A     PD 29
      Placebo    CR 13
      Placebo    PR 51
      Placebo    SD 34
      Placebo    PD 17
      ;
   run;
   proc sql noprint;
      select sum(count) into: trt1 from adpsa where trt01p='Drug_A';
      select sum(count) into: trt2 from adpsa where trt01p='Placebo';
   quit;
   %let trt1=&trt1; %let trt2=&trt2;
   %put trt1=&trt1, trt2=&trt2;

  data cat;
    set adpsa;
    if trt01p='Drug_A' then percent = round(count/&trt1*100, .1);
    if trt01p='Placebo' then percent = round(count/&trt2*100, .1);
   run;
  proc sgplot data=cat;
       vbar trt01p/group=aval response=percent groupdisplay=cluster
grouporder=data
        dataskin=pressed attrid=aval datalabel=count;
        xaxis display=(nolabel noticks);
        yaxis values=(0 to 100 by 20) label='Percentage (%)';
        keylegend/title='Best Overall Response'
        title = 'Bar Graph for Best Overall Response';
        inset "P-value= &pval"/position=topright;
        inset "Difference in ORR (95% CI): &df" /position=topleft;
        inset "           95% CI for ORR: &ORR1 " /position=left;
        inset "95% CI for ORR: &ORR2   " /position=right;
        *&df, &pval, &ORR1, &ORR2 correspond to the statistics in Figure 3*;
   run;
```

Mean Standard Error Plot

The plot of mean change over time is commonly used in conjunction with standard error bars. In a mean standard error plot, the X-axis represents time and the Y-axis represents the overall mean. Each point on the graph represents the overall mean value of the data at a specific time point and is associated with an error bar. The bar above the point is computed by adding one standard error while the bar below the point is computed by subtracting one standard error. The mean plots visualize the shift in mean over time and the error bars display the overall distribution of the data.

Figure 12 is an example of Mean Standard Error plot

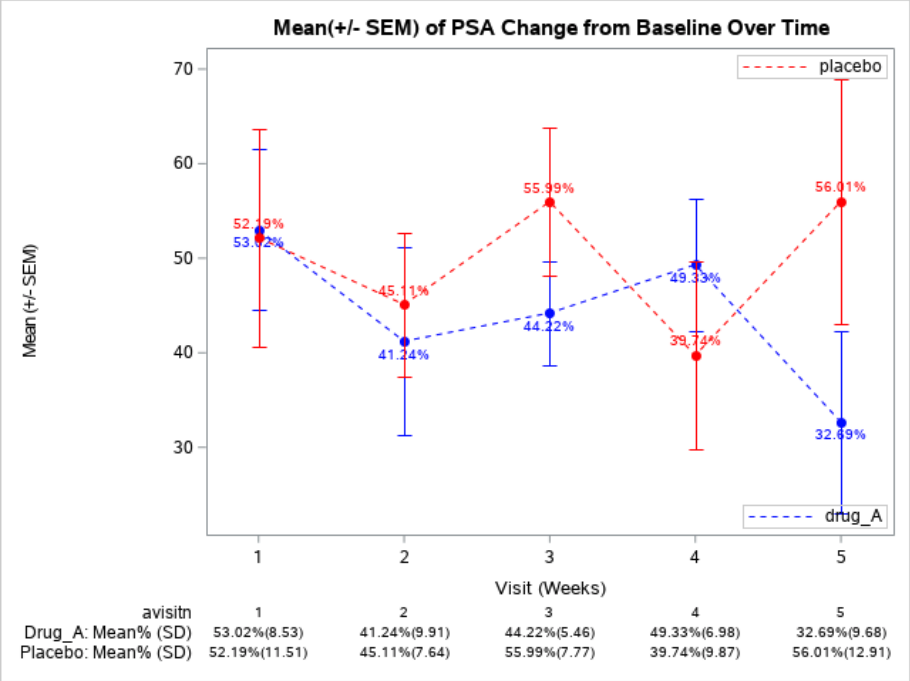

Figure 12 Mean Standard Error Plot of Drug vs. Placebo

```
/* Create sample data */
data adpsa(drop=i);
length trt01p $100;
   do trt01p='drug_A','placebo';
         do i=1 to 10;
               do avisitn=1 to 5;
                     psapchg=ranuni(0)*100;
                     output;
               end;
         end;
end;
proc sort;by trt01p avisitn;run;

/* Calculate the mean and standard error for each treatment */
 %macro meanout(trt=, out=);
 proc means data=adpsa noprint;
   where trt01p = "&trt";
   by trt01p avisitn;
   var psapchg;
   output out=meanout (drop=_type_ _freq_)
   mean=&out._mean stderr=&out._stderr;
 run;
 data &out.;
   set meanout;
   &out._ll = &out._mean - &out._stderr;
   &out._ul = &out._mean + &out._stderr;
 proc sort;by avisitn;
 run;
 %mend;
```

```
%meanout(trt=drug_A, out=trt1);
%meanout(trt=placebo, out=trt2);


data master;
merge trt1(drop=trt01p) trt2(drop=trt01p);
by avisitn;
seq + 1;
if trt1_mean ne '' then
trt1_msd = strip(put(trt1_mean,
8.2))||'%('||strip(put(trt1_stderr,8.2))||')';
if trt2_mean ne '' then
trt2_msd = strip(put(trt2_mean,
8.2))||'%('||strip(put(trt2_stderr,8.2))||')';
trt1_meanp = strip(put(trt1_mean,8.2))||'%';
trt2_meanp = strip(put(trt2_mean,8.2))||'%';
run;

proc sgplot data=master noautolegend;
  series x=avisitn y=trt1_mean/lineattrs=(color=blue pattern=2)
  datalabel=trt1_meanp datalabelpos=bottom datalabelattrs=(color=blue)
  name = 'drug_A' legendlabel='drug_A';
  series x=avisitn y=trt2_mean/lineattrs=(color=red pattern=2)
  datalabel=trt2_meanp datalabelpos=top datalabelattrs=(color=red)
  name = 'placebo' legendlabel='placebo';
  scatter x=avisitn y=trt1_mean/yerrorlower=trt1_ll yerrorupper=trt1_ul
                               markerattrs=(color=blue symbol=CircleFilled)
                               errorbarattrs=(color=blue);
  scatter x=avisitn y=trt2_mean/yerrorlower=trt2_ll yerrorupper=trt2_ul
                               markerattrs=(color=red symbol=CircleFilled)
                               errorbarattrs=(color=red);
  xaxistable avisitn/x=seq location=outside;
  xaxistable trt1_msd/x=seq location=outside label='Drug_A: Mean% (SD)';
  xaxistable trt2_msd/x=seq location=outside label='Placebo: Mean% (SD)';
  yaxis label='Mean (+/- SEM)';
  xaxis label='Visit (Weeks)';
  title1 'Mean(+/- SEM) of PSA Change from Baseline Over Time';
  keylegend 'drug_A'/location=inside position=bottomright;
  keylegend 'placebo'/location=inside position=topright;
run;
```

Spaghetti Plot

A spaghetti plot looks like a plate of spaghetti. It's a line plot displaying the trend for each individual subject. Each line in a spaghetti plot represents the change in value for one patient over multiple visits. The example code below shows how to generate a PSA spaghetti plot using the SGPLOT procedure with the dummy data.

Figure 13 is an example of spaghetti plot.

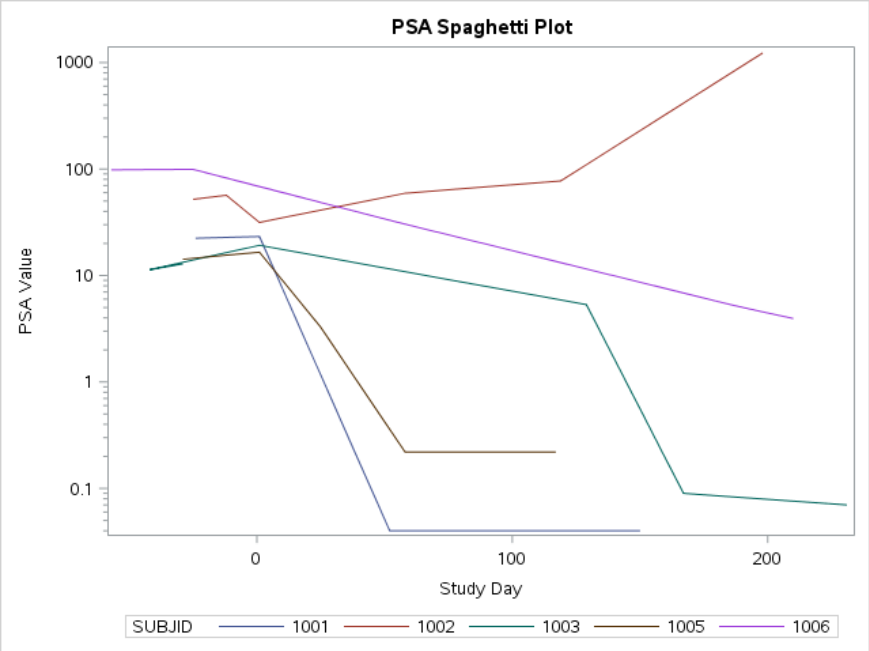

Figure 13 PSA Spaghetti Plot

```
data adpsa;
format SUBJID z4. ;
input  SUBJID ADY AVAL TRT01PN;
PARAMCD = 'PSA';
datalines;
   1001 -24  22.41    1
   1001 1    23.22    1
   1001 52   0.04     1
   1001 150  0.04     1
   1002 -25  52.01    1
   1002 -12  56.78    1
   1002 1    31.52    1
   1002 58   59.21    1
   1002 119  77.39    1
   1002 198  1230.34  1
   1003 -29  12.85    1
   1003 -42  11.34    1
   1003 1    19.21    1
   1003 129  5.33     1
   1003 167  0.09     1
   1003 231  0.07     1
   1005 -29  14.21    1
   1005 1    16.54    1
   1005 25   3.26     1
   1005 58   0.22     1
   1005 117  0.22     1
   1006 -57  98.23    1
   1006 -25  99.22    1
   1006 54   32.11    1
   1006 187  5.21     1
```

```
   1006 210  3.94     1
   ;
run;

proc sgplot data=adpsa(where=(trt01pn=1));
Title 'PSA Spaghetti Plot';
series x=ady y=aval/group=subjid;
yaxis label='PSA Value' type=log logbase=10 logstyle=logexpand minor;
xaxis label='Study Day';
run;
```

Forest Plot

Forest plots are commonly used to visualize statistical results for subgroup analysis in randomized controlled trials. The general structure of a forest plot consists of three parts: subgroup labels, plots of hazard ratios with associated confidence intervals, and relevant summary statistics.

The relevant statistics are summarized in the statistics panel in the plot. The statistics can be, for instance, number of patients and number of events for treatment versus placebo, hazard ratio with corresponding confidence interval, and p-value. It's important to note that in the plot panel of a forest plot the dots represent treatment effects, for instance, measured by hazard ratios in the Figure 14 below. Hazard ratios typically display as dots or diamond along with the corresponding confidence intervals: the amount of variation for the estimates of hazard ratio. The vertical line (x=1.0) in the middle indicates a hazard ratio of 0, which suggests no treatment effects. With placebo as reference group, a hazard ratio less than one indicates favoring treatment over placebo and the dot falls into the left-hand side of the plot panel. When validating forest plots, programmers need to cross-check with summary tables and assure that the statistics in the plot aligns with the statistics in the tables.

The example code below illustrates how to pre-summarize the statistics before creating a forest plot and how to create a plot with the SGPLOT procedure based on the pre-summarized data.

Figure 14 is an example of forest plot.

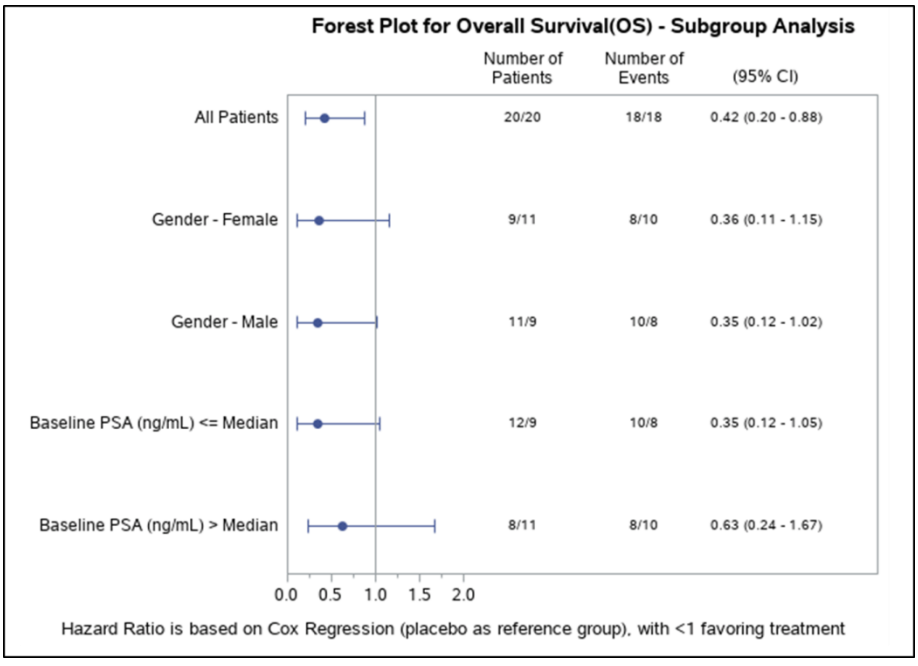

Figure 14 Forest Plot for Subgroup Analysis

```
data adeff;
   input aval cnsr trtpn Sex $ tglscor psablmgr $ @@;
   datalines;
   179   0   1   F   6   Y   378   1   1   M   6   Y
   256   0   1   F   9   Y   355   0   1   M   5   Y
   262   0   1   M   7   N   319   0   1   M   8   N
   256   0   1   F   7   Y   256   0   1   M   9   N
   255   0   1   M   8   N   171   0   1   F   7   Y
   224   1   1   F   9   Y   325   0   1   M   8   N
   (more data here…)
   ;
run;


%macro m_forestplot(indata= , llabel= , orderid = , var=);
   ods graphics on;
   ods output Quartiles=_quart_&var CensoredSummary =_cs_&var;
         proc lifetest data=&indata plots=none;
         time aval*cnsr(1);
         strata trtpn ;
         run;
   ods output close;
   ods graphics off;

   *Get Number of Patients*;
   data cs_&var (keep=trtpn npatients );
         set _cs_&var(where=(trtpn in (1,2)));
         length npatients $25;
         npatients =strip(put(total,best.));
   run;
   proc transpose data=cs_&var out=_tcs_&var prefix=_;
         var npatients;
         id trtpn;
   run;
   data tcs_&var (keep=orderid col0 col1);
         set _tcs_&var;
         length col0 col1 $200;
         orderid = input(&orderid,best.);
         col0= &llabel.;
         col1=catx('/', _1,_2);
   run;
   *Get Number of Events*;
   data ecs_&var (keep=trtpn nevents );
         set _cs_&var(where=(trtpn in (1,2)));
         length nevents $25;
         nevents =strip(put(failed,best.));
   run;
   proc transpose data=ecs_&var out=_etcs_&var prefix=_;
   var nevents;
   id trtpn;
   run;
   data etcs_&var (keep= orderid col0 col2 );
   set _etcs_&var;
   length col0 col2 $200;
   orderid = input(&orderid,best.);
   col0= &llabel.;
   col2=catx('/', _1,_2);
```

```
   run;

   * Hazard ratio (based on Cox Regression);
   ods output ParameterEstimates=_hzdata_&var;
   proc phreg data=&indata;
   class trtpn;
   model aval*cnsr(1)=trtpn / ties=discrete risklimit;
   run;

   data hzdata_&var (keep= orderid col0 or lcl ucl col3 HazardRatio
   HRLowerCL HRUpperCL);
   set _hzdata_&var;
   length col0 col3 $200;
   orderid = input(&orderid,best.);
   col0= &llabel.;
   or=HazardRatio;
   lcl=HRLowerCL;
   ucl=HRUpperCL;
   col3=strip(put(HazardRatio,8.2))||'('||strip(put(HRLowerCL,8.2))||'-
   '||strip(put(HRUpperCL,8.2))||')';
   run;

   data final_&var;
         merge tcs_&var etcs_&var hzdata_&var;
         by orderid;
   run;
%mend m_forestplot;

%m_forestplot(indata=adeff,llabel='All Patients',orderid=1, var=allsubj);

%m_forestplot(indata=adeff(where=(sex='F')),llabel='Gender - Female',
             orderid=2,var=sexf);
%m_forestplot(indata=adeff(where=(sex='M')),llabel='Gender - Male',
             orderid=3,var=sexm);

%m_forestplot(indata=adeff(where=(psablmgr='Y')),
             llabel='Baseline PSA (ng/mL) <= Median',
             orderid=4,var=psablmgry);
%m_forestplot(indata=adeff(where=(psablmgr='N')),
             llabel='Baseline PSA (ng/mL) > Median',
             orderid=5,var=psablmgrn);

data master;
set final_allsubj final_sexf final_sexm final_psablmgry final_psablmgrn;
run;
data master;
   set master(keep=orderid col0 col1 col2 col3
                HazardRatio HRLowerCL HRUpperCL);
   length col11 col12 col13 $30;
   col11='Number of Patients';
   col12='Number of Events';
   col13='(95% CI)';
proc sort;by descending orderid;
run;
proc sgplot data=master noautolegend;
   title 'Forest Plot for Overall Survival(OS) - Subgroup Analysis';
   scatter y=col0 x=HazardRatio/xerrorupper=HRUpperCL
```

```
            xerrorlower=HRLowerCL markerattrs=(symbol=circlefilled);

    /*--Display statistics columns on X2 axis--*/
    scatter y=col0 x=col11 / markerchar=col1 x2axis;
    scatter y=col0 x=col12 / markerchar=col2 x2axis;
    scatter y=col0 x=col13 / markerchar=col3 x2axis;

    refline 1 / axis=x;
    refline 1/ axis=x lineattrs=(pattern=shortdash) transparency=0.5;
    *inset 'Favors Treatment'  / position=bottomleft;
    *inset 'Favors Placebo'  / position=bottom;
    xaxis  offsetmin=0 offsetmax=0.70 min=0 max=2 minor display=(nolabel);
    x2axis offsetmin=0.4 display=(noticks nolabel);
    yaxis display=(noticks nolabel) offsetmin=.1 offsetmax=.05;
    footnote 'Hazard Ratio is based on Cox Regression (placebo as reference
    group), with <1 favoring treatment';
run;
```

CONCLUSION

Statisticians and statistical programmers work collaboratively to perform statistical analyses, prepare specifications to build analysis datasets, and report the analytical results. An understanding of survival analysis and the therapeutic guidelines (including tumor-related data, RECIST, oncology endpoints, special censoring and confirmation handling rules) is required for a statistical programmer to understand the inferential analyses in statistical analysis plans (SAP) and mock-up shells in greater depth. As a result, statistical programmers can transform oncology data into useful and meaningful analysis data structures, and create and validate the results more efficiently.

REFERENCES

[1] Halabi, S., & Kelly, W. K. (2010). Oncology Clinical Trials. Demos Medical Publishing.

[2] P Therasse et al, "New response evaluation criteria in solid tumors: Revised RECIST guideline (version 1.1)," European Journal of Clinical Oncology, pp. 45: 228-247, 2009.

[3] Yao, B. (2016). Data and Analysis Considerations in Oncology Clinical Trials. Biometrics & Biostatistics International Journal, 4(4). doi:10.15406/bbij.2016.04.00101

[4] ALLISON, P. D. (2018). SURVIVAL ANALYSIS USING SAS: A practical guide, second edition. S.l.: SAS INSTITUTE.

CONTACT INFORMATION

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

Joy Zeng
Pfizer Inc
joy.zengjy@gmail.com
https://github.com/joyzengyaya/