PharmaSUG 2023 - Paper SA-068

Validating novel maraca plots – R and SAS love story

Nicole Major, Srivathsa Ravikiran, Monika Huhn, Samvel B. Gasparyan, Martin Karpefors, AstraZeneca

ABSTRACT

Hierarchical composite endpoints (HCE) are complex endpoints combining outcomes of different types and different clinical importance into an ordinal outcome that prioritizes the clinically most important (e.g., most severe) event of a patient. HCE can be analyzed with the win odds, an adaptation of the win ratio to include ties. One of the difficulties in interpreting HCE is the lack of proper tools for visualizing the treatment effect captured by HCE, given the complex nature of the endpoint. The recently introduced maraca plot solves this issue by providing a comprehensive visualization that clearly shows the treatment effects on the HCE and its components. The *maraca* package in the R software provides an easy-to-use implementation of maraca plots, building on powerful features provided by the *ggplot2* package. The *maraca* package also provides the calculations for the complex statistical analyses involved in deriving the maraca plot, including the overall treatment effect characterized by win odds. An important consideration for using maraca plots in regulatory setting is the question of how to validate the analyses involved in deriving the maraca plots. In this paper an approach is demonstrated using the SAS® software to validate the outputs generated by the R *maraca* package and thereby combining the best of two worlds: the flexible plotting capabilities of the R software and the powerful data manipulation and statistical analysis tools of the SAS® software.

INTRODUCTION

Hierarchical composite endpoints (HCE) are complex endpoints combining outcomes of different types and different clinical importance into an ordinal outcome that prioritizes the clinically most important (e.g., most severe) event of a patient (Gasparyan, Buenconsejo et al. 2022). Most commonly, HCE combine multiple time-to-event outcomes with a single continuous outcome. For example, in a fixed follow-up clinical trial, patients may be followed for death or hospitalization, with death being the worst outcome. If two patients experience death, then the timing of death can be used to distinguish the patient with a worse outcome (a later death being a better outcome). If the given patient did not experience death before end of the follow-up, then they will contribute to the analysis with the hospitalization they experienced, with a later hospitalization being better. Otherwise, if the patient is alive and without hospitalization, then a laboratory value can be used in the analysis, for example, a patient with a higher change from baseline may be considered as having a better outcome than the patient with a lower change from baseline in this laboratory value. In this case, the HCE consists of two time-to-event (TTE) outcomes and one continuous outcome. HCE can be analyzed using win odds (Gasparyan, Kowalewski et al. 2021), an adaptation of win ratio to include ties (Gasparyan, Folkvaljon et al. 2021). The number of TTE outcomes can be arbitrary. For visualization of HCE the maraca plot was introduced (Karpefors, Lindholm et al. 2022).

The maraca plot (see Figure 1) combines the features of a Kaplan–Meier plot to show the cumulative percentage of patients with given TTE outcome (TTE1, TTE2, and so on) over fixed follow-up period in each treatment group, with a violin plot (with nested box plot) showing the median and density of the continuous distribution. Therefore, the left part of the plot shows clinically severe outcomes that the treatment intends to prevent, and hence, the separation of curves would be indicative of treatment effect if the treatment has a lower cumulative percentage of patients with outcomes TTE1, TTE2, and so on, compared to the control group. The x-axis for the continuous outcomes corresponds to its unit and a beneficial treatment effect is characterized by a shift to the right. In the maraca plot, the distribution of the continuous outcome is visualized with a horizontal violin plot and a nested box plot. The associated vertical dashed line shows the median value for the continuous outcome. In addition, since each patient contributes to the HCE only once, the width of each category (time-to-event outcomes and continuous outcome) on the maraca plot corresponds to the percentage of that category in the overall population, mirroring the contribution of components to the composite.

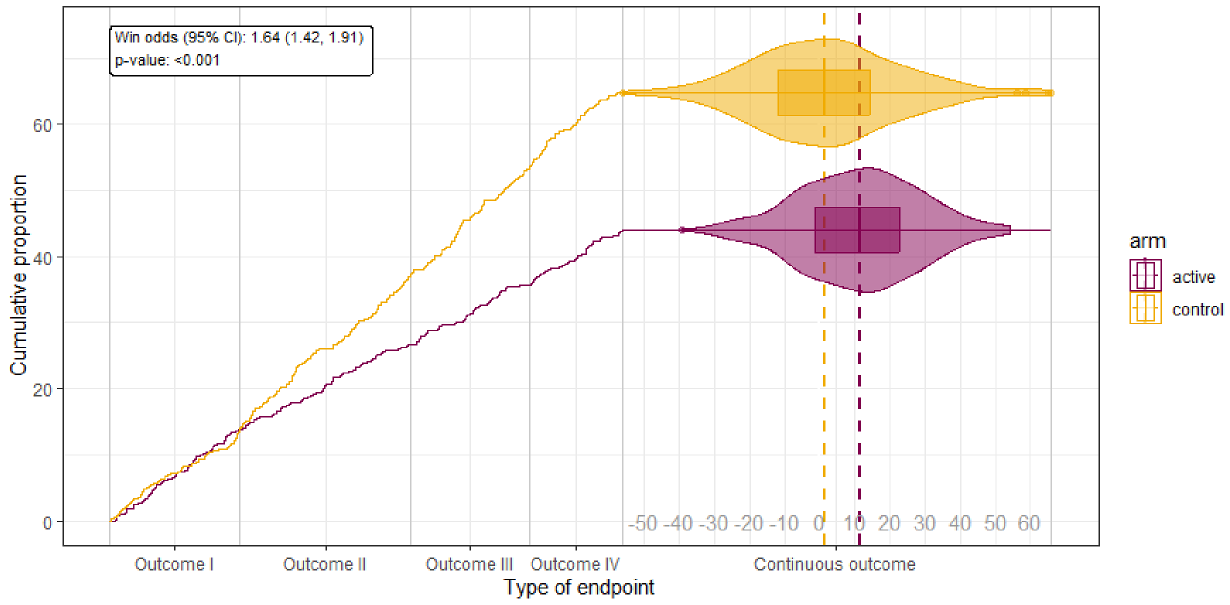
If maraca plots are used to visualize clinical study results for regulatory submissions, the results should be double programmed for validation purposes. Below a validation approach using the SAS® software is provided. The adopted validation approach is the following. First, after plotting the maraca in R, a special function in the *maraca* package is used, called *validate_maraca_plot(),* to extract the data sets associated with the plot. These data sets are then validated using the SAS® software, by deriving the statistics included in these data sets independently from the input data set using SAS® software procedures. Finally, comparison is done of the validation data sets with the data sets extracted from the maraca plot using the SAS® procedure PROC COMPARE. The SAS® software version 9.4 and R software version 4.2.2 are used.

MARACA PLOTS IN R

The maraca plot is implemented in the *maraca* package (Karpefors and Gasparyan 2022) of the R software (R Core Team 2022). The *maraca* package is based on *ggplot2* graphical objects (Wickham 2016). Below the built-in data set *hce_scenario_a* from the *maraca* package is used to create an example of the maraca plot (see the documentation of the *maraca()* function in the R software).

```
library(maraca)
data(hce_scenario_a, package = "maraca")
data <- hce_scenario_a

column_names <- c(outcome = "GROUP", arm = "TRTP", value = "AVAL0")
tte_outcomes <- c("Outcome I", "Outcome II", "Outcome III", "Outcome IV")
continuous_outcome <- "Continuous outcome"
arm_levels <- c(active = "Active", control = "Control")

maraca_object <- maraca(
  data, tte_outcomes, continuous_outcome, arm_levels, column_names,
  fixed_followup = 3*365, compute_win_odds = TRUE
)

AZ_colors <- c("#830051", "#F0AB00")

plot(maraca_object, density_plot_type = "default") + theme_bw() +
  scale_color_manual(values = AZ_colors) +
  scale_fill_manual(values = AZ_colors)
```

Figure 1 - An example of a maraca plot



In addition to the default type of the maraca plot shown in the figure above, which is a nested combination of violin and box plot, it is possible to create maraca plots that depict the continuous distribution as violin plots by specifying the argument density_plot_type as “violin” for only violin plots, “box” for only boxplots, “scatter” for scatter plots.

It is also notable and very convenient that since the *plot(marca_object, …)* creates a *ggplot* object all powerful features of *ggplot2* objects are preserved. For example, themes can be easily changed, or the settings can easily be used to expand and customize the plot according to the grammar of graphics’ (*ggplot2*) conventions. In the example, a black-and-white theme was applied, and the treatment arm colors were changed to AstraZeneca corporate colors.

VALIDATION OUTPUT FROM R

The *maraca* package contains a function customized for validation of maraca visualizations. The idea behind this functionality is to extract key metrics from the *ggplot* object created using the maraca plot function. These allow then for automated checks on the plot output compared to, for example, versions of the plot created in other programming languages.
Running the *validate_maraca_plot()* function on a maraca plot object, with the default (nested box plot/violin plot) density setting for the continuous variable, will output a list of 5 data sets that represent data in the maraca plot corresponding to,

1) the proportions of the HCE components (`proportions`),
2) the time-to-event data (`tte_data`),
3) the boxplot statistics (`boxstat_data`),
4) the violin distribution data (`violin_data`) and
5) the win odds statistics (`wo_stats`).

The list can be converted to a set of comma-separated value (CSV) files that can be used in the SAS® software as described in the next section of this paper.

```r
library(tibble)
library(dplyr)
library(tidyr)

path <- "" #The path of the CSV files

maraca_plot <- plot(maraca_object, density_plot_type = "default")
validation_list <- validate_maraca_plot(maraca_plot)

as.data.frame(validation_list$proportions) %>%
  rownames_to_column(., "outcome") %>%
  rename("proportion" = "validation_list$proportions") %>%
  write.csv(., file = paste0(path,"proportions.csv"), row.names = FALSE)

write.csv(validation_list$tte_data, file = paste0(path,"tte_data.csv"), row.names = FALSE)

stat_data <- validation_list$boxstat_data %>%
  as_tibble() %>%
  unnest_wider(outliers, names_sep = "") %>%
  pivot_longer(., cols = -group, names_to = "stat_name", values_to = "values") %>%
  filter(!is.na(values))
write.csv(stat_data, file = paste0(path,"boxplot_data_long.csv"), row.names = FALSE)

validation_list$violin_data %>%
  select(group, x, density) %>%
  write.csv(., file = paste0(path,"violin_data.csv"), row.names = FALSE)

write.csv(t(validation_list$wo_stats), file = paste0(path,"wo_stats.csv"), row.names = FALSE)
```

VALIDATION OF THE OUTPUT USING SAS

The following validation approach is followed:

- The validation was carried out on a simulated dataset *hce_scenario_a*.
- Data sets from the *maraca* package are generated using the *validate_maraca_plot()* function noted above. In the scope of validation, the data sets for proportions, time-to-event curve, boxplot, and violin plot are included.
- SAS®-generated data sets were compared with R results using the PROC COMPARE procedure. Due to the computational differences in the procedures between R & SAS® in the density estimation used for the violin plots (PROC KDE in SAS® and *geom_violin()* in R) the precision is set to 1E-3 across all compared data sets and compared values were found to be matching at this precision. To more closely match the density estimations across the two languages, other options in PROC KDE and *geom_violin()* should be explored.

The function *validate_maraca_plot()* provides also a validation data set including the win odds and its confidence interval that are used in the legend of the maraca plot (see Figure 1). The calculation of the win odds and its confidence interval in the maraca package is done through the package *hce* (Gasparyan 2022) and therefore its validation is out of scope for this paper.

DATA

This validation was carried out on the *hce_scenario_a* dataset, which is a simulated dataset included in the *maraca* package. The data frame has 1,000 observations with even split between the active and control arms. There are seven variables in this data set:

- X: the row number of the observation.
- SUBJID: the patient's identifier.
- GROUP: the most severe clinical outcome the patient underwent.

- GROUPN: a numerical value for GROUP. This variable isn't required for computation.
- AVAL0: the time-to-event data for the hard outcomes and the continuous data for the continuous outcome.
- AVAL: an ordered sequence of values where the AVAL0 value associated with the patient is offset by GROUPN. This variable isn't required for computation.
- TRTP: the patient's treatment group.

The input variables of interest for this validation are GROUP, AVAL0, and TRTP.

VALIDATION STEPS

The validation of *validate_maraca_plot()* outputs is carried out in the following order: the proportions, time-to-event curve, boxplot, and violin plot. The proportions dataset must be validated first as it's used to derive variables needed for the visualization components.

PROPORTIONS

The proportions of all outcomes are visualized as vertical lines and are used for defining the widths of each component on the plot. Therefore, the outcome with the highest proportion will have the highest width on the plot. This is an important feature in the interpretation of the hierarchical composite endpoint.

The SAS® software procedure used to validate the proportions is PROC FREQ. The code below calculates the proportions based on the input data set, which can be compared with the validation data set generated from the function *validate_maraca_plot()*.

```
proc freq data = hce_scenario_a noprint;
      tables GROUP*GROUPN / out = proportions_sas;
run;
```

TIME-TO-EVENT CURVE

The data set for the time-to-event (herein referred to as TTE) curve produced by the *maraca* package is called `tte_data` and has three variables: *group*, *x*, and *y*. The *y* variable is the cumulative proportion of patients that undergo a clinical event over the timeframe defined by the column *x*. Plotting the columns *x* versus *y* by the treatment group (the *group* variable) will produce the first part of the maraca plot with Kaplan-Meier curves.

To validate the columns *x* and *y* the following input variables are needed: the time variable for the TTE outcomes that is used in the cumulative distribution function (which relies on input variable *AVAL0*), the length of fixed follow-up, and the type of the TTE outcome (the *GROUP* variable).

First, variable "I" must be derived where I = 1 for "Outcome I", I = 2 for "Outcome II" and so on with I = 5 for "Continuous outcome". Essentially, this variable is like the existing GROUPN variable, but it instead ascends sequentially from 1 to the max number of outcomes in the data.

The macro variable *fixed_followup* contains the length of fixed follow-up in the study and is set at the beginning of the program.

With these three variables (*I*, *fixed_followup*, and *AVAL0*), the transformed time variable TIME_CDF can be computed, which uses sequentially the timings of the TTE outcomes 1, 2, and so on.

The variable *x* from the validation dataset has a range from 0 to 100. Therefore, the variable TIME_CDF should be transformed so that each outcome has the width corresponding to its overall proportion to get the variable *x.*

To do so, the variables AVAL0, fixed_followup, PROPORTION, and LAG_CP variables are needed. The first two variables already exist, and the PROPORTION variable has been derived in the `proportion_sas` data set created in the previous section. LAG_CP requires the derivation of a cumulative proportion variable based on the outcome. Once this has been determined, LAG_CP uses the *lag()* function on the calculated cumulative proportion. For this example, the proportion of Outcome I is 13.8 and its LAG_CP is 0 because there is no outcome that is more severe than Outcome I. For Outcome II, the proportion is 18.1, its' cumulative proportion is 31.9 (13.8 from Outcome I + 18.1 from Outcome II). The LAG_CP of Outcome II is 13.8 because that's the cumulative proportion of all the more severe outcomes than itself.

```
data tte2;
   set tte1;

   * create variable to be used in CDF;
   if GROUP = "Continuous outcome" then TIME_CDF = (I + 1) * &fixed_followup;
   else TIME_CDF = AVAL0 + &fixed_followup * (I - 1);

   * create variable to be used in validation against maraca plot;
   X = LAG_CP + AVAL0 / &fixed_followup * PROPORTION;
run;
```

Now PROC UNIVARIATE is used to produce the cumulative distribution of TIME_CDF by each treatment group:

```
proc univariate data = tte2 noprint;
   by TRTP;
   cdfplot / vscale = PERCENT;
   var TIME_CDF;
   ods output CDFPlot = cdf;
run;
```

There are a couple items to note about the code above. TTE1, the predecessor of TTE2, has previously been sorted on TRTP, so there's no need to use the NOTSORTED option here. As the *y* variable produced by the *maraca* package is in the form of a percentage, vscale = PERCENT is the option to use for the PROC UNIVARIATE call.

At this point, the notable information the `cdf` dataset contains is the treatment information, ECDFX, and ECDFY. ECDFY is equivalent to the *y* variable created by the *maraca* package, so there's no further computation to do with this variable. However, as noted above, the ECDFX variable is not the variable the maraca package uses for its plotting of the TTE curve. That variable is the X variable that was created in the data step that creates `tte2`. As the GROUP variable hasn't been retained by the `cdf` dataset, in order to retrieve these X values, a merge must be performed between `cdf` and `tte2` on the treatment column and the ECDFX and TIME_CDF variables for `cdf` and `tte2` respectively.

After this merge is performed, only those with ECDFY values > 0 and < 100 are retained for comparison. This eliminates the observations with GROUP = "Continuous outcome" as well as the observations where ECDFX and ECDFY are both equal to 0.

CONTINUOUS OUTCOME

The following two sections discuss visualizing the continuous outcome of the hierarchical composite endpoint. In the "default" plot type, this produces the boxplot and violin plot. Prior to delving into those sections, it must be noted that the maraca plot transforms the value of the time-to-event variable to a different scale used for plotting purposes. This transformed variable is stored in variable X below:

```
data continuous;
   set hce_scenario_a(rename = (GROUP   = OUTCOME)
                      where  = (OUTCOME = "Continuous outcome"));
   GROUP = lowcase(TRTP);

   X = (100 - &prop) * (AVAL0 - &min) / (&max - &min) + &prop;
run;
```

The three macro variables above that are crucial to determining the value of X for each observation are derived in previous steps. *&prop* is the proportion of the data that has GROUP = "Continuous outcome", which in this case is 54.4. *&min* is the lowest AVAL0 occurring in the data while *&max* is the highest AVAL0 occurring in the data, approximately –55.324 and 66.041 respectively.

With this X variable created, the boxplot and violin plot validation can now be created.

Boxplot

PROC SGPLOT was used to create the boxplot statistics that are used to validate the boxplot created in the maraca package.

```
proc sgplot data = continuous;
   ods output sgplot = boxplot(drop = X GROUP);
   hbox X / category = GROUP;
run;
```

The default quantile determination between R and SAS® software are different. The SAS® software uses the empirical distribution with averaging while R makes use of a linear interpolation algorithm when needed (see, for example, the R documentation for the function *quantile())* (Hyndman and Fan 1996). There are linear interpolation algorithms for determining the quantiles available in SAS®, but not the same version used in R by default. This required creating code that use the same logic as R.

The following formulas are implemented in the SAS® software to match the R software quantiles (for the full code refer to the supplement). First h must be calculated. h is the index of the observation to choose for the quantile:

$$h = (N - 1)p + 1$$

where N is the number of observations and p is the desired quantile expressed as a decimal. If h is an integer, then x_h will be the quantile, where x_h is the h^{th} observation after sorting the observations in ascending order by X. However, if h is not an integer, then a linear interpolation algorithm is used to determine the quantile Q_p:

$$Q_p = x_{\lfloor h \rfloor} + (h - \lfloor h \rfloor)(x_{\lceil h \rceil} - x_{\lfloor h \rfloor})$$

where $x_{\lfloor h \rfloor}$ corresponds to floor(x_h) in the SAS® software, $\lfloor h \rfloor$ to floor(h), and $x_{\lceil h \rceil}$ to ceil(x_h).

The code that implements the first, second, and third quartiles outputs a separate dataset called `quartiles`. This is appended to the data set that contains the remaining boxplot statistics. There is some updating to the names of the statistics in the SAS® software as they differ from the output data sets provided by the *validate_maraca_plot()* function. After concatenating the two data sets and formatting the resulting combined data set, it is ready for PROC COMPARE.

Violin Plot

PROC KDE produces the density estimates that are needed to validate the violin plot from the *maraca* package. Some adjustments must be made to the procedure in order to match R's defaults. The code to do so is below:

```
proc kde data = continuous;
   by GROUP;
   univar X / plots =(density) method = SROT NG = 512 truncate;
   ods output DensityPlot = violin_sas(rename = (DensityX = x) drop = VarName);
run;
```

The "truncate" option is used to "trim the tails" of the violin plot. It sets the lowest grid limit to the smallest value for X and the highest grid limit to the largest value of X, thereby "trimming the tails", which is what R's *geom_violin()* does (Wickham 2016). Setting NG = 512 is done to, once again, align to R's default. This parameter is used to determine the number of grid points to use for each kernel density estimate.

The default method for determining the bandwidth for the SAS® software is the Sheather-Jones plug-in method (Sheather 2004). When looking into the *geom_density()* documentation, which computes the kernel density estimates for the violin plot, the default is "nrd0" (Wickham 2016). "nrd0" is Silverman's rule of thumb "unless the quartiles coincide when a positive result will be guaranteed" (see the R software documentation for the function *bw.nrd0()*). To match this as closely as possible to the methods available in SAS® software, the method variable is set to "SROT".

Using this combination of parameters yielded a comparison that matched with the criterion set to 1E-3. This was determined to be sufficient for the purposes of this validation, but more investigation into PROC KDE may result in achieving a match with a greater level of precision.

CONCLUSION

Recently, novel maraca plots were introduced as a method for visualization of hierarchical composite endpoints in clinical trials. The maraca plots are implemented in the R package *maraca* based on the powerful tools provided by *the ggplot2* package. The *maraca* package also contains a function to extract validation datasets that provide all the information used in constructing maraca plots. If maraca plots are used to visualize clinical study results for regulatory submissions, the results should be double programmed for validation purposes. This paper provides the validation of the data sets associated with the maraca plot in the SAS® software. This validation approach combines the best of two worlds - the flexible plotting capabilities of R and the powerful data manipulation and statistical analysis tools of the SAS® software, and thus makes the use of maraca plots in regulatory submissions easily adoptable.

REFERENCES

Gasparyan, S. B. (2022). "hce: Design and Analysis of Hierarchical Composite Endpoints." R package version >=0.5.0. https://CRAN.R-project.org/package=hce.
Gasparyan, S. B., J. Buenconsejo, E. K. Kowalewski, J. Oscarsson, O. F. Bengtsson, R. Esterline, G. G. Koch, O. Berwanger and M. N. Kosiborod (2022). "Design and Analysis of Studies Based on Hierarchical Composite Endpoints: Insights from the DARE-19 Trial." Ther Innov Regul Sci **56**(5): 785-794. https://doi.org/710.1007/s43441-43022-00420-43441.
Gasparyan, S. B., F. Folkvaljon, O. Bengtsson, J. Buenconsejo and G. G. Koch (2021). "Adjusted win ratio with stratification: calculation methods and interpretation." Statistical Methods in Medical Research **30**(2): 580-611. https://doi.org/510.1177/0962280220942558.
Gasparyan, S. B., E. K. Kowalewski, F. Folkvaljon, O. Bengtsson, J. Buenconsejo, J. Adler and G. G. Koch (2021). "Power and sample size calculation for the win odds test: application to an ordinal endpoint in COVID-19 trials." Journal of Biopharmaceutical Statistics **31**(6): 765-787. https://doi.org/710.1080/10543406.10542021.11968893.
Hyndman, R. J. and Y. Fan (1996). "Sample quantiles in statistical packages." The American Statistician **50**(4): 361-365.
Karpefors, M. and S. B. Gasparyan (2022). "maraca: The Maraca Plot: Visualization of Hierarchical Composite Endpoints in Clinical Trials." R package version >=0.5.0. https://CRAN.R-project.org/package=maraca.
Karpefors, M., D. Lindholm and S. B. Gasparyan (2022). "The maraca plot: A novel visualization of hierarchical composite endpoints." Clinical Trials **20**(1): 84-88. https://doi.org/10.1177/17407745221134949.
R Core Team (2022). R: A language and environment for statistical computing. Version 4.2.2. Vienna, Austria, R Foundation for Statistical Computing. https://www.R-project.org/.
Sheather, S. J. (2004). "Density estimation." Statistical Science **19**(4): 588-597.
Wickham, H. (2016). ggplot2: Elegant Graphics for Data Analysis, Springer New York, NY. https://doi.org/10.1007/978-0-387-98141-3.

CONTACT INFORMATION

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

Nicole Amai Major
AstraZeneca Pharmaceuticals
nicole.major@astrazeneca.com

Srivathsa Ravikiran
AstraZeneca Pharmaceuticals
Srivatha.ravikiran@astrazeneca.com

Monika Huhn
AstraZeneca Pharmaceuticals
monika.huhn@astrazeneca.com

Samvel Gasparyan
AstraZeneca Pharmaceuticals
samvel.gasparyan@astrazeneca.com

Martin Karpefors
AstraZeneca Pharmaceuticals
martin.karpefors@astrazeneca.com

APPENDIX

```
proc datasets library=work mt=data kill noprint;run;
/*********************
*       Set up       *
*********************/
* Set up macrovariables;
options validvarname=v7;
%let data_area = ;
%let output = ;
%let fixed_followup = 1095;

/*********************
*     Load Data      *
*********************/;
* Simulated data;
proc import datafile = "&data_area/hce_scenario_a.csv"
         out = hce_scenario_a (keep = SUBJID GROUP GROUPN AVAL0 AVAL TRTP)
         DBMS  = CSV;
run;

* Remove formats;
proc datasets lib = work memtype = data;
         modify hce_scenario_a;
                  attrib _all_ format = ;
run;

* Proportions created by validate_maraca();
proc import datafile = "&data_area/proportions.csv"
         out  = proportions_r
         DBMS = CSV;
run;

* TTE curve created by validate_maraca();
proc import datafile = "&data_area/tte_data.csv"
         out  = tte_r
         DBMS = CSV;
run;

* Boxplot statistics created by validate_maraca();
proc import datafile = "&data_area/boxplot_data_long.csv"
         out  = boxplot_r
         DBMS = CSV
         replace;
run;

* Violin plot created by validate_maraca();
proc import datafile = "&data_area/violin_data.csv"
         out  = violin_r(keep = group x density)
         DBMS = CSV;
         guessingrows = max;
run;

/*********************
*    Proportions     *
*********************/;
* Create proportions;
proc freq data = hce_scenario_a noprint;
         tables GROUP*GROUPN / out = proportions_sas;
run;
```

```
proc sort data = proportions_sas(rename = (GROUP = OUTCOME PERCENT = PROPORTION)) out =
proportions_sas(keep = OUTCOME PROPORTION);
         by GROUPN;
run;

/*************
*    TTE     *
*************/;
* Create proportions needed for TTE derivation;
data proportions;
         set proportions_sas;

         CUMULATIVE_PROPORTION + PROPORTION;

         LAG_CP = lag(CUMULATIVE_PROPORTION);
         if LAG_CP = . then LAG_CP = 0;
run;

proc sql;
         * Gather distinct outcomes;
         create table groups as
         select distinct hsa.GROUP, hsa.GROUPN, proportions.PROPORTION, proportions.LAG_CP
         from hce_scenario_a as hsa
                   left join
                   proportions
                   on hce_scenario_a.GROUP = proportions.OUTCOME
         order by GROUPN;

         * Create numeric ordering variable for each outcome;
         create table groups1 as
         select groups.*, monotonic() as I
         from groups;

         * Merge derived data onto primary dataset;
         create table tte1 as
         select hce_scenario_a.*, groups1.I, groups1.PROPORTION, groups1.LAG_CP
         from hce_scenario_a
                   left join
                   groups1
                   on hce_scenario_a.GROUP = groups1.GROUP
         order by TRTP, GROUPN, AVAL0;
quit;

data tte2;
   set tte1;

   * Create variable to be used in CDF;
   if GROUP = "Continuous outcome" then TIME_CDF = (I + 1) * &fixed_followup;
   else TIME_CDF = AVAL0 + &fixed_followup * (I - 1);

   * Create variable to be used in validation against maraca plot;
   X = LAG_CP + AVAL0 / &fixed_followup * PROPORTION;
run;

proc univariate data = tte2 noprint;
         by TRTP;
         cdfplot / vscale = PERCENT;
         var TIME_CDF;
         ods output CDFPlot = cdf;
run;
```

```
proc sql;
	* Merge derived X variable onto cdf dataset;
	create table tte3 as
	select cdf.*, tte2.GROUP as OUTCOME, tte2.X
	from cdf
		left join
		tte2
		on cdf.TRTP = tte2.TRTP and cdf.ECDFX = tte2.TIME_CDF;
quit;

* Format for comparison;
data tte4;
	set tte3(where = (ECDFY > 0 and ECDFY < 100)
	                   drop  = VarName CDFPlot);

	GROUP  = lowcase(TRTP);
	Y      = ECDFY;

	keep GROUP X Y;
run;

* Order variables and sort observations;
proc sql;
	create table tte_sas as
	select GROUP, X, Y
	from tte4
	order by Y, GROUP;
quit;

/*********************
*  Continuous Data   *
*********************/;
proc sql noprint;
	* Get min and max of AVAL0 in order to create plotting scale;
	select min(AVAL0) format = best32., max(AVAL0) format = best32.
	into : min trimmed,
	     : max trimmed
	from hce_scenario_a(where = (GROUP = "Continuous outcome"));

	* Determine cumulative proportion of discrete outcomes in order to create plotting
scale;
	select cumulative_proportion
	into : prop trimmed
	from proportions(where = (OUTCOME ^= "Continuous outcome"))
	having CUMULATIVE_PROPORTION = max(CUMULATIVE_PROPORTION);
quit;

data continuous;
	set hce_scenario_a(rename = (GROUP   = OUTCOME)
	                                        where  = (OUTCOME = "Continuous outcome"));

	GROUP = lowcase(TRTP);

	* Transform to plotting scale for validation;
	X = (100 - &prop) * (AVAL0 - &min) / (&max - &min) + &prop;
run;

proc sort data = continuous;
	by TRTP;
run;
```

```
/*****************
*    Boxplot     *
*****************/;
proc sgplot data = continuous;
         ods output sgplot = boxplot(drop = X GROUP);
         hbox X / category = GROUP;
run;

* Rename what comes out of PROC SGPLOT for simplicity;
proc sql noprint;
         select catx("=", name, label)
         into : rename_list separated by " "
         from dictionary.columns
         where libname = "WORK" and lowcase(MEMNAME) = "boxplot" and label ^= "";
quit;

* Keep statistics needed for comparison;
data boxplot1(where = (STATISTIC ^in("N", "STD", "MEAN", "Q1", "MEDIAN", "Q3")));
         set boxplot(rename = (&rename_list)
                                    where = (STAT ^= ""));

         STATISTIC = STAT;
         VALUES    = X;

         keep GROUP STATISTIC VALUES;
run;

* Start quartiles derivation;
data quartiles;
         P = .25;
         output;

         P = .50;
         output;

         P = .75;
         output;
run;

proc sort data = continuous out = cont1(keep = TRTP X);
         by TRTP X;
run;

* Create index variable;
data cont2;
         set cont1;
         by TRTP;

         if first.TRTP then I = 0;
         I + 1;
run;

proc sql;
         * Determine N;
         create table arms1 as
         select distinct TRTP, count(*) as N
         from cont2
         group by TRTP;

         * Add needed information to derive quantiles;
         create table arms2 as
         select arms1.*, quartiles.p, (N - 1) * P + 1 as H, floor(calculated H) as H_FLOOR,
ceil(calculated H) as H_CEIL
         from arms1,
                   quartiles;

         * Find observations based on index for floor(h);
         create table floor as
         select arms2.*, cont2.X as X_FLOOR
         from arms2
                   left join
                   cont2
                   on arms2.TRTP = cont2.TRTP and arms2.H_FLOOR = cont2.I;

         * Find observations based on index for ceil(h);
         create table ceil as
         select arms2.*, cont2.X as X_CEIL
         from arms2
                   left join
                   cont2
                   on arms2.TRTP = cont2.TRTP and arms2.H_CEIL = cont2.I;

         * Merge floor and ceiling values onto main data;
         create table arms3 as
         select floor.*, ceil.X_CEIL
         from floor
                   left join
                   ceil
                   on floor.TRTP = ceil.TRTP and floor.P = ceil.P;
quit;

* Calculate quartiles;
data box_quartiles;
         set arms3;

         GROUP  = lowcase(TRTP);

         if      p = .50 then STATISTIC = "MEDIAN";
         else if p = .25 then STATISTIC = "Q1";
         else if p = .75 then STATISTIC = "Q3";

         VALUES = X_FLOOR + (H - H_FLOOR) * (X_CEIL - X_FLOOR);

         keep GROUP STATISTIC VALUES;
run;
* End quartiles derivation;

* Format dataset for comparison;
data boxplot2;
         set boxplot1 box_quartiles;
         length STAT_NAME $ 15;

         if      STATISTIC = "DATAMIN" then STAT_NAME = "x_lowest";
         else if STATISTIC = "MIN"     then STAT_NAME = "whisker_lower";
         else if STATISTIC = "Q1"      then STAT_NAME = "hinge_lower";
         else if STATISTIC = "MEDIAN"  then STAT_NAME = lowcase(STATISTIC);
         else if STATISTIC = "Q3"      then STAT_NAME = "hinge_upper";
         else if STATISTIC = "MAX"     then STAT_NAME = "whisker_upper";
         else if STATISTIC = "DATAMAX" then STAT_NAME = "x_highest";

         if      STAT_NAME = "x_lowest"      then STAT_ORD = 1;
         else if STAT_NAME = "whisker_lower" then STAT_ORD = 2;
         else if STAT_NAME = "hinge_lower"   then STAT_ORD = 3;
         else if STAT_NAME = "median"        then STAT_ORD = 4;
         else if STAT_NAME = "hinge_upper"   then STAT_ORD = 5;
         else if STAT_NAME = "whisker_upper" then STAT_ORD = 6;
         else if STAT_NAME = "x_highest"     then STAT_ORD = 7;
run;

proc sort data = boxplot2;
         by GROUP STAT_ORD VALUES;
run;

* Final formatting;
data boxplot3;
         set boxplot2;
         by GROUP STAT_ORD VALUES;

         if STATISTIC = "OUTLIER" then do;
                  if first.GROUP then I = 0;

                  I + 1;
                  STAT_NAME = "outliers" || strip(put(I, best.));
                  STAT_ORD  = 8;
         end;
run;

* Sorting and ordering variables for final dataset;
proc sql;
         create table boxplot4 as
         select GROUP, STAT_NAME, VALUES, STAT_ORD
         from boxplot3
         order by GROUP, STAT_ORD, STAT_NAME;

         create table boxplot_sas as
         select GROUP, STAT_NAME, VALUES
         from boxplot4;
quit;

/*********************
*    Violin Plot     *
*********************/;
proc kde data = continuous;
         by GROUP;
         univar X / plots =(density) method = SROT NG = 512 truncate;
         ods output DensityPlot = violin_sas(rename = (DensityX = x) drop = VarName);
run;

* Remove attributes before comparing;
proc datasets lib = work memtype = data;
         modify proportions_r;
                  attrib _all_ format   = ;
                  attrib _all_ informat = ;
                  attrib _all_ label    = "";

         modify proportions_sas;
                  attrib _all_ label    = "";
                  attrib _all_ informat = ;

         modify tte_r;
                  attrib _all_ format   = ;
                  attrib _all_ informat = ;
         modify boxplot_r;
                  attrib _all_ format   = ;
                  attrib _all_ informat = ;
                  attrib _all_ label    = "";

         modify boxplot_sas;
                  attrib _all_ label = "";

         modify violin_r;
                  attrib _all_ format   = ;
                  attrib _all_ informat = ;

         modify violin_sas;
                  attrib _all_ label = "";
run;

* Comparisons;
proc compare base = proportions_r compare = proportions_sas method = absolute criterion = 1E-
3;
run;

proc compare base = tte_r compare = tte_sas method = absolute criterion = 1E-3;
run;

proc compare base = boxplot_r compare = boxplot_sas method = absolute criterion = 1E-3;
run;

proc compare base = violin_r compare = violin_sas method = absolute criterion = 1E-3;
run;
```