PharmaSUG 2025 - Paper DV-297

Swanky Sankey Enhancements: Transforming a Graph with Pretty Curves to a Research Tool uncovering Deeper Scientific Insights

Siqi Wang, Arcsine Analytics

Toshio Kimura, Arcsine Analytics

Abstract

Sankey plots show the flow of data from one state to another over time. Existing implementations are available displaying the main bar graph with curvy paths flowing from one category to another[1]. However, reviewers are left wondering how many patients are going to and coming from each category. Additionally, the relative spacing on the x-axis representing time is not maintained; therefore, 4-week and 12-week intervals are represented with equal spacing.

We propose the following enhancements to improve upon these shortcomings. First, we will introduce sidebars along with the option to display n and percent for the number of patients going to and coming from each category. This will proactively address the most asked question from clinical and medical writing colleagues. Second, we will use relative spacing for the x-axis representing time so that unequal time intervals will be appropriately displayed. For example, a 4-week interval will be one third the space of a 12-week interval. Third, a summary table showing all the values will be generated. This table can be used to determine the value for any part of the Sankey plot which the medical writers can use within a report or publication.

These improvements address the most widely cited shortcomings of the current Sankey implementation and will facilitate a wider adoption of the Sankey plot. These enhancements greatly improve interpretability and will transform the Sankey plot from a visually appealing graph with pretty curvy lines to an informative figure that provides researchers with deeper scientific insight.

Introduction

In clinical data visualization, it is important to show how patients move through different response levels across different visits to identify the general trend over time. Sankey diagram serves as a valuable tool for data analysis, providing critical insights into patient pathways.

A typical Sankey plot consists of cumulative bar plots showing the distribution of patients in different levels at each time point, along with curves connecting the barploting representing the transitions of patients between consecutive time points (Figure 1).

[1] S.Rosanbalm, Getting Sankey with Bar Charts, PharmaSUG 2015

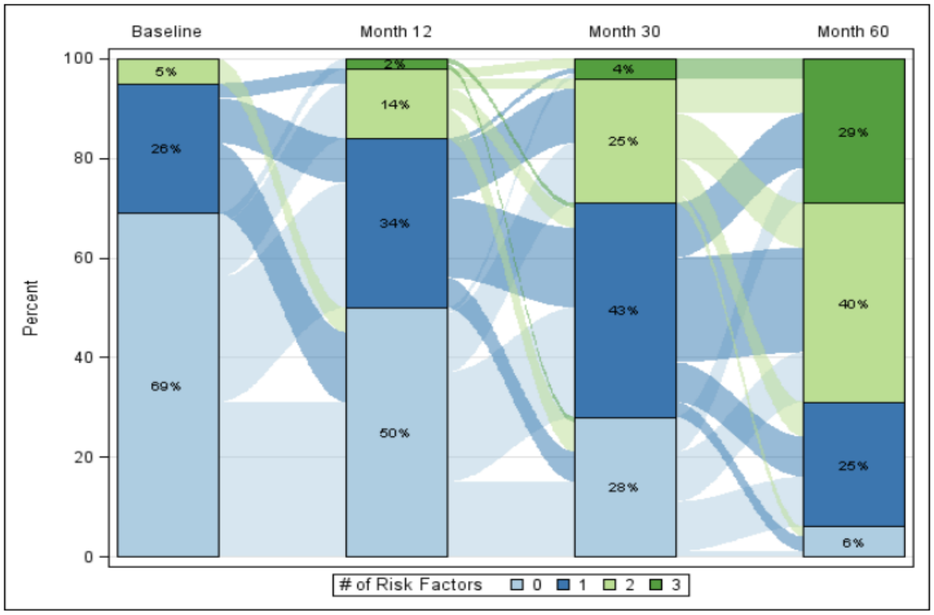


Figure 1: Typical Sankey Plot[1]

The traditional Sankey plot is visually appealing and an improvement over standard bar plots. However, upon further inspection, it lacks the depth of information needed for clinical interpretation. There are 3 main shortcomings:

1) Number of subjects at each time point: The previous implementation does not display the number of subjects at a given time point.

2) Number of subjects transitioning: The previous implementation also does not detail the number of patients transitioning from one category to another. The primary advantage of the Sankey plot over a standard bar plot is to display the flow of patients from one category to another. While this is visually displayed in the Sankey plot, the traditional implementation does not quantify this flow.

3) Time interval: The spacing between timepoints is also uniform, which misrepresents the actual timeline and can be misleading when interpreting patient progress over uneven intervals.

To overcome the 3 main limitations of the traditional Sankey plot as described above, we redesigned the visualization to be more informative, accurate to the data, and aligned with the practical needs of clinical analysis.

1) Number of subjects at each time point: The enhanced Sankey Plot will display the number of patients as well as the percentage at each time point for each category level. This enhancement directly addresses the lack of quantitative context in the original plot.

2) Number of subjects transitioning: The enhanced Sankey Plot will introduce sidebars alongside each bar of the Sankey Plot. These sidebars display both inflow and outflow distributions, showing both

the n and percent, detailing a clear breakdown of how each group is composed and where the patients came from and where patients are heading to. This significantly improves interpretability, making the plot information dense. This also allows reviewers to immediately understand the size of each flow, making the transitions more meaningful and grounded in the data

To further supplement the sidebars and quantify the number of patients transitioning, we will also optionally output a table to document the frequencies at each time point, providing an alternative format to facilitate report writing (Table 1).

3) Time interval: Lastly, we addressed the issue of misleading timepoint spacing. In traditional Sankey implementation, timepoints are spaced equally regardless of actual intervals. Our updated design spaces timepoints proportionally to the true differences in time. This ensures that the flow of patients is accurately represented in relation to the study timeline, clearly representing the visits spacing.

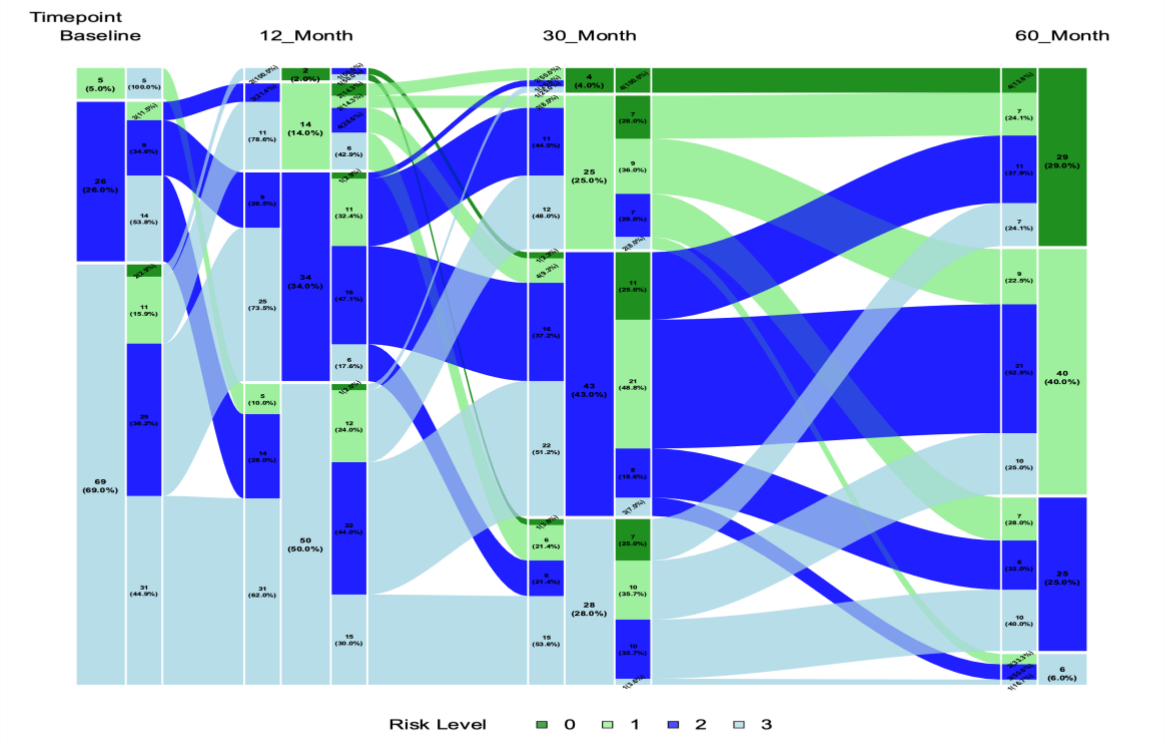
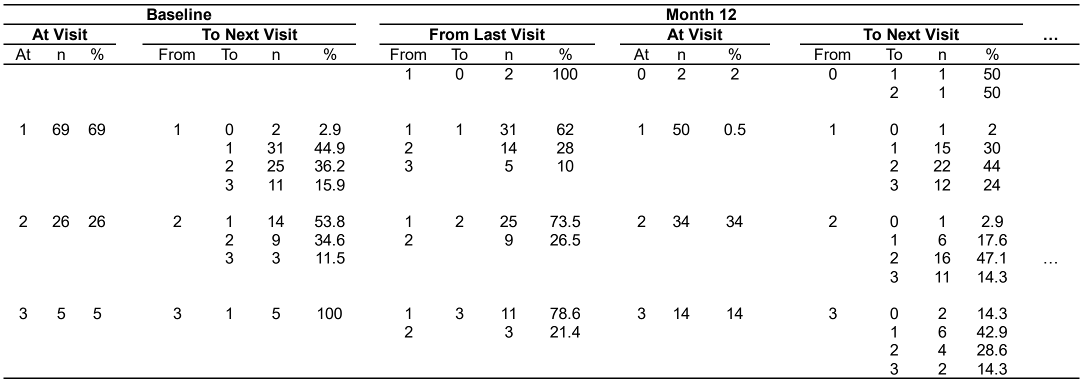


Figure 2: Enhanced Sankey Plot



Baseline							Month 12											
At Visit			To Next Visit				From Last Visit				At Visit			To Next Visit				…
At	n	%	From	To	n	%	From	To	n	%	At	n	%	From	To	n	%	
							1	0	2	100	0	2	2	0	1	1	50	
															2	1	50	
1	69	69	1	0	2	2.9	1	1	31	62	1	50	0.5	1	0	1	2	
				1	31	44.9	2		14	28					1	15	30	
				2	25	36.2	3		5	10					2	22	44	
				3	11	15.9									3	12	24	
2	26	26	2	1	14	53.8	1	2	25	73.5	2	34	34	2	0	1	2.9	
				2	9	34.6	2		9	26.5					1	6	17.6	
				3	3	11.5									2	16	47.1	…
															3	11	14.3	
3	5	5	3	1	5	100	1	3	11	78.6	3	14	14	3	0	2	14.3	
							2		3	21.4					1	6	42.9	
															2	4	28.6	
															3	2	14.3	

Table 1: Table Output (showing first 2 visits)

Enhanced Sankey Plot Implementation

Number of subjects at each time point

To reflect the number of subjects at each timepoint, we added the statistical output to each main rectangle and the sidebar. Once the label `text_pct_row` is calculated, X and Y coordinates will be calculated to make sure the text is centered on each rectangle. Users can also customize which statistics are displayed on the plot by specifying `show_value=`, with options including `N`, `PERCENT`, or both (`N_PERCENT`, which is the default).

```
/* Create text for the percentage row */

      %if %upcase(%sysfunc(strip(&show_value))) = N_PERCENT %then %do;

            text_pct_row = catx('0A'x, COUNT, cats('(', put(PCT_ROW,
            5.1),'%)'));

      %end;

      %else %if %upcase(%sysfunc(strip(&show_value))) = N %then %do;

            text_pct_row = catx('0A'x, COUNT);

      %end;

      %else %if %upcase(%sysfunc(strip(&show_value))) = PERCENT %then %do;

            text_pct_row = catx('0A'x, cats('(', put(PCT_ROW, 5.1), '%)'));

      %end;

/* Center the label horizontally and vertically */

      label_x = last_x + &barwidth / 2;

      label_y = y - ((100 - 5) / 100) * pct_row / 2;
```

Number of subjects transitioning

One of the key innovations in our enhanced Sankey plot is the addition of sidebars—thin bars placed to the left and right of each main node, rendered in differentiated colors. These sidebars provide immediate visual context by showing how many patients are entering or exiting a specific category. This is achieved by identifying transitions across timepoints: after intaking the raw dataset, the macro constructs a series of derived frequency tables that track patient-level flows from one category to another at each timepoint. For every small bar, the macro calculates the vertical space each flow should occupy in the sidebar based on the proportion of patients moving from or to a specific group. Each of these transitions is then rendered as a small rectangle, defined by `start_x1, start_x2, start_y1`, and `start_y2` (for inflow) and similarly by `end_x1, end_x2, end_y1`, and `end_y2` (for outflow). These rectangles are stored in the `_edge_rectangles` dataset, which holds all necessary coordinates and metadata.

```
data _edge_rectangles;

set _paths2;

/* Hold cumulative totals for group counts */

    array start_n {&n_visits, &n_grps} (%sysevalf(&n_visits * &n_grps) * 0);

    array end_n {&n_visits, &n_grps} (%sysevalf(&n_visits * &n_grps) * 0);

/* Calculate Y-axis for start and end rectangles */

    start_y1 = start_group_min + start_group_diff * (start_n(starting_node,
    start_index) / start_group_n);

    start_y2 = start_y1 + start_group_diff * (n_move / start_group_n);

      end_y1 = end_group_min + end_group_diff * (end_n(ending_node,

      end_index) / end_group_n);

    end_y2 = end_y1 + end_group_diff * (n_move / end_group_n);

/* Calculate X-axis for start and end rectangles */

      start_x1 = start_group_x - 3.8;

      start_x2 = start_group_x - 0.2;

      end_x1 = end_group_x + 0.2;

      end_x2 = end_group_x + 3.8;

/* Update cumulative totals for stacking */

    start_n(starting_node, start_index) + n_move;

    end_n(ending_node, end_index) + n_move;
```

Each sidebar block is also color-coded according to the group it represents, providing a clear visual indication of where patients are coming from and where they are heading. This detailed visual layering makes the flow dynamics much easier to interpret, especially in complex visualization with multiple response categories.

Time interval

The macro calculates the relative spacing between timepoints using the actual values passed through the visit's parameter. Once the unique timepoints are identified, the macro determines the minimum (`min_week`) and maximum (`max_week`) values to compute the full time range. Each timepoint is then assigned a normalized ratio based on its position within this range. These ratios are used to scale the x-axis coordinates of all boxes, ensuring that the spacing between nodes accurately reflects the real intervals between timepoints.

```
/* Step 1: Get number of unique visit values    */
proc sql noprint;
    %local n_visits i null;
    /* Count number of unique visit values */
    select count(distinct visits) into :n_visits
    from _frq;
    select distinct visits format=12. into :node1-
    from _frq;
    %do i = 1 %to &n_visits;
        %local n_group&i;
    %end;
    /* Count distinct groups within each visit */
    select visits, count(distinct group)
        into :null, :n_group1-
    from _frq
    group by visits;
quit;
/* Step 2: Build mapping between week & visit   */
proc sql;
    create table unique_weeks as
    select distinct timepoint as week, visits
    from _temp
    order by week;
quit;
/* Step 3: Calculate min, max, and range        */
proc sql noprint;
    select min(week) into :min_week from unique_weeks;
    select max(week) into :max_week from unique_weeks;
quit;
```

```
%let range = %sysevalf(&max_week - &min_week);
/* Step 4: Assign normalized ratio to each week */
data week_ratios;
    set unique_weeks;
    retain min_week max_week range;
    min_week = &min_week;
    max_week = &max_week;
    range = &range;
    /* Ratio is a normalized [0,1] value + shift to avoid 0 */
    ratio = 1 + (week - min_week) / range;
run;
```

Other Enhancements

The enhanced Sankey plot not only adds dense information to the visualization but also provides users with greater flexibility and control over how the plot is displayed. Users can choose whether to include missing values using the `keepnull=` parameter. If set to `YES`, missing values will be treated as a separate group (defaulting to gray) and displayed at the top of each bar. If set to `NO`, missing values will be excluded, and the space will be fully allocated to non-null categories. Below, Figure 3 shows that the data initially contains some missing values, and 5% of the data drops to null at each timepoint.

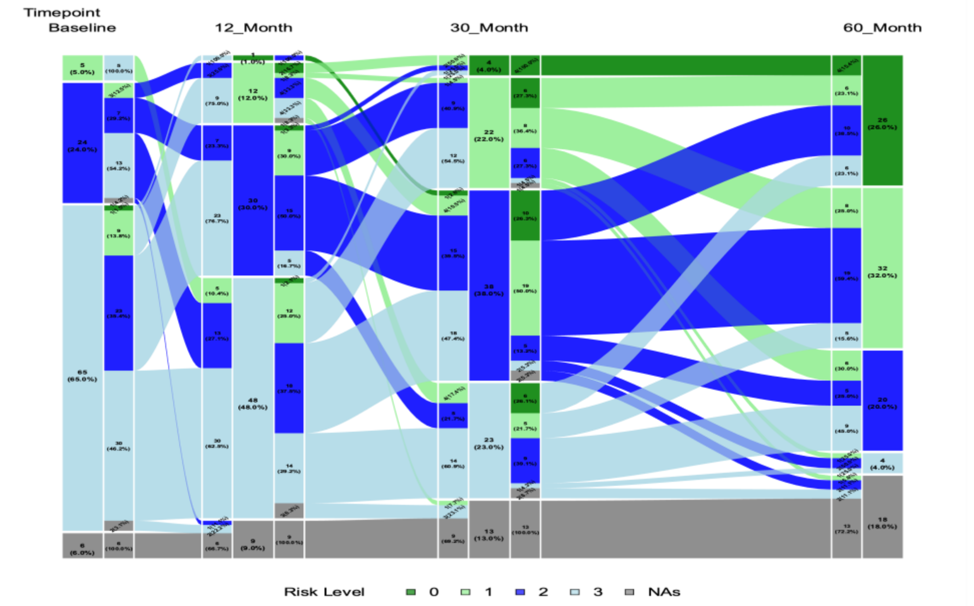


Figure 3: Sankey Plot with Missing Data

Additionally, users can tailor the color scheme using the `colors=` parameter. They can assign specific colors to each group, with a default palette of 10 preset colors, where gray is used to represent null or missing values. Output format is customizable as well through the `filetype=` parameter, supporting formats like svg or png depending on the intended use case. Users can also turn the sidebars on or off with `sidebar=`. These customization options give users significant freedom to tailor the plot to their needs, making the macro both powerful and adaptable. Figure 4 illustrates a version with different output colors (`colors = grey lightpink lightpurple pink lightred`), with `sidebar = NO` and `show_value = N`. This provides a more general representation, suitable when the user wants to display only high-level information.

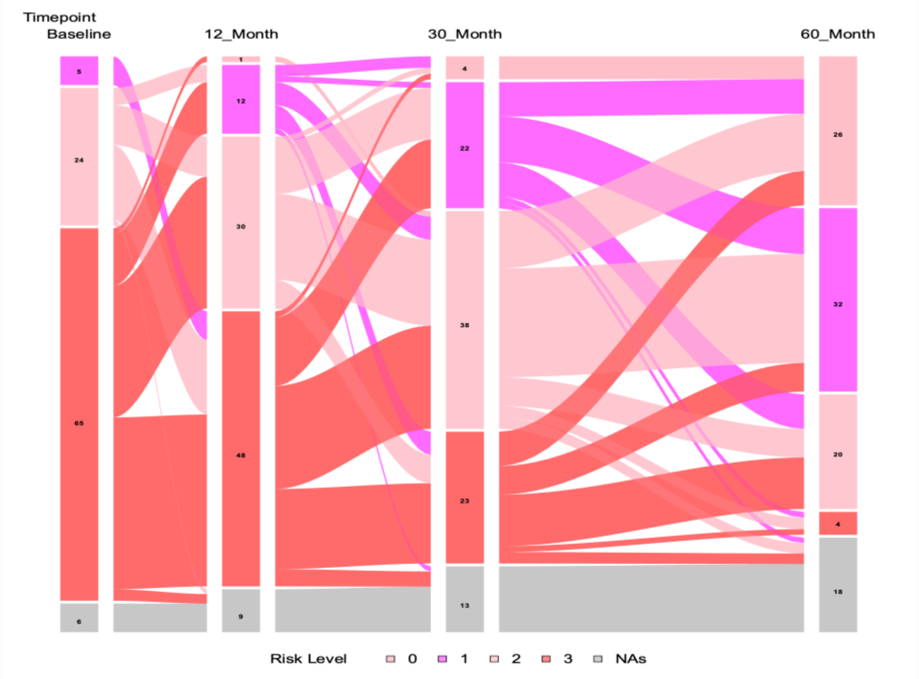


Figure 4: Sankey Plot with Sidebar Turned off

Checkpoints are built throughout the macro to ensure robust handling of various customization options. If a user provides an unsupported or invalid input, the macro issues informative messages to guide correction. Additionally, when rendering the final plot, the macro automatically adjusts the angle of statistical labels when they might overlap eachothers. Smaller group's statsitics will be rotated 45 degrees to avoid overlapping. These built-in measures make the macro more stable, robost, and user-friendly.

Sankey Macro Implementation

Macro Parameters

Parameter	Core	Definition	Notes/Example
data	Required	Input dataset to be used in the macro.	Must contain columns for id, visits, and group.
id	Required	Unique identifier for subjects.	Default: USUBJID.
visits	Required	Numeric timepoints.	Default: AWTARGET.
respvar	Required	Levels of outcomes or categories for subjects.	Default: AVAL.

sidebar	Optional	Determines whether the sidebar is included in the plot.	Valid values: YES or NO. Default: YES.
keepnull	Optional	Determines whether null (missing) values are included in the plot.	Valid values: YES or NO. Default: YES.
show_value	Optional	Choose the statistics that is being displayed	Valid values: PERCENT and N, separated by space Default: N PERCENT
colors	Optional	Choose the colors for each respvar level	Valid values:1(red) 2(orange) null(grey) Default: 10 preset colors with grey representing null.
filetype	Optional	Choose the output formart for the plot	Valid: png or svg Default: svg
barwidth	Optional	Controls the width of the main bars in the plot.	Default: 4.
bargap	Optional	Percentage of space on y-axis between group categories.	Default: 0.5.
curve_rectangle_gap	Optional	Gap as a percentage between the ends of curves and the nodes.	Default: 3.

FUTURE IMPROVEMENTS

One potential improvement is to better support “fan-out” patterns, where individuals drop out and are visualized as fading out at their dropout time—beyond just being classified as missing (e.g., “NA”). Another area of improvement could be filtering out groups that are too small, so the plot doesn’t become cluttered with excessively thin lines. To make the plot more clinically informational, patients showing transitions across two or more risk categories in adjacent visits could be highlighted to emphasize significant changes.

Discussion

The enhanced Sankey macro we developed is one of the most comprehensive and flexible implementations. It addresses many of the limitations seen in traditional Sankey plots, such as equal time spacing and lack of quantitative data. Key features include proportional timepoint spacing, inflow/outflow sidebars, and dynamic statistical annotations. Data visualization with dense information like this enhanced Sankey macro can improve patient management and support the identification of meaningful trends.

REFERENCES

Rosanbalm, S., Getting Sankey with Bar Charts. (2015)

Contact Information

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

Name: Siqi Wang

Company: Arcsine Analytics

E-mail: amber.wang@arcsineanalytics.com

Name: Toshio Kimura

Company: Arcsine Analytics

E-mail: toshio.kimura@arcsineanalytics.com