PharmaSUG 2023 - Paper AP-268

Display Layout Specification to Flexibly Design and Generate Tables

Songgu Xie, Regeneron Pharmaceutical, Inc.
Michael Pannucci, Independent;
Toshio Kimura, Independent

ABSTRACT

Currently, tables are created in SAS by manually processing the data and defining PROC REPORT to generate a predefined fixed table. If different layouts are required, extensive data processing will be required to create alternative table layouts. We propose a new process to decouple the statistical analysis procedure from the display layout generation process. The first step is performing the statistical procedure and storing results in standard results datasets (RDS), similar to CDISC ARS. RDS will be presented separately, thus not in this paper's scope. The second step is processing RDS to generate displays. We propose a new specification called the display layout specifications (DLS) to drive display generation. Through DLS, we will flexibly design various table layouts. DLS will be an input into a flexible display generation macro that will process RDS according to DLS to greatly facilitate the creation of alternative table layouts.

INTRODUCTION

COMMON SCENARIO

Tables constitute essential components of clinical reports. Statistical programmers frequently face a typical scenario in their daily work, where they receive requests from statisticians or medical directors to create various table layouts to meet diverse requirements. These requests may encompass a broad range of topics, including clinical outcomes, safety, and efficacy measures. For example, in a given case, there might be an initial request to create an ANCOVA table, including summary statistics across visits and ANCOVA-based statistics. The rows should display visits and statistics, while the columns should present treatments.

				PBO (reference) (N=xxx)	Trt 1 (N=xxx)	Trt 2 (N=xxx)
Baseline	Summary Statistics		N	xxx	xxx	xxx
			n	xxx	xxx	xxx
			Mean	xx.x	xx.x	xx.x
			SE	x.x	x.x	x.x
Week 4	Summary Statistics		N	xxx	xxx	xxx
			n	xxx	xxx	xxx
			Mean	xx.x	xx.x	xx.x
			SE	x.x	x.x	x.x
	Statistics based on ANCOVA	LS Means	LS mean	xx.x	xx.x	xx.x
			SE	x.x	x.x	x.x
			95% CI	xx.x, xx.x	xx.x, xx.x	xx.x, xx.x
		Difference vs Reference	LS mean difference		xx.x	xx.x
			SE		x.x	x.x
			95% CI		xx.x, xx.x	xx.x, xx.x
			P-value		0.xxxx	0.xxxx
Week 8	Summary Statistics		N	xxx	xxx	xxx
			n	xxx	xxx	xxx
			Mean	xx.x	xx.x	xx.x
			SE	x.x	x.x	x.x
	Statistics based on ANCOVA	LS Means	LS mean	xx.x	xx.x	xx.x
			SE	x.x	x.x	x.x
			95% CI	xx.x, xx.x	xx.x, xx.x	xx.x, xx.x
		Difference vs Reference	LS mean difference		xx.x	xx.x
			SE		x.x	x.x
			95% CI		xx.x, xx.x	xx.x, xx.x
			P-value		0.xxxx	0.xxxx

Figure 1. Table Layout #1

Nonetheless, during the analysis, new requirements from stakeholders may emerge, possibly requesting additional statistics in columns to better illustrate the analysis results or make different comparisons. Generating various table layouts is a routine task for statistical programmers. Provided the data values are available, transforming the dataset and utilizing the `REPORT` procedure in SAS to create tables manually is not exceedingly difficult.

						Statistics based on ANCOVA						
		Summary Statistics				LS Means			Difference vs Reference			
Visit	Treatment	N	n	Mean	SE	Est.	SE	95% CI	Est.	SE	95% CI	P value
Baseline	PBO (reference)	xxx	xxx	xx.x	x.x							
	Trt 1	xxx	xxx	xx.x	x.x							
	Trt 2	xxx	xxx	xx.x	x.x							
Week 4	PBO (reference)	xxx	xxx	xx.x	x.x	xx.x	x.x	xx.x, xx.x				
	Trt 1	xxx	xxx	xx.x	x.x	xx.x	x.x	xx.x, xx.x	xx.x	x.x	xx.x, xx.x	0.xxxx
	Trt 2	xxx	xxx	xx.x	x.x	xx.x	x.x	xx.x, xx.x	xx.x	x.x	xx.x, xx.x	0.xxxx
Week 8	PBO (reference)	xxx	xxx	xx.x	x.x	xx.x	x.x	xx.x, xx.x				
	Trt 1	xxx	xxx	xx.x	x.x	xx.x	x.x	xx.x, xx.x	xx.x	x.x	xx.x, xx.x	0.xxxx
	Trt 2	xxx	xxx	xx.x	x.x	xx.x	x.x	xx.x, xx.x	xx.x	x.x	xx.x, xx.x	0.xxxx

Figure 2. Table Layout #2

		Visit (Unit)		
		Baseline	4	8
PBO (reference) (N=xxx)	n	xx	xx	xx
	LS Mean	xx.x	xx.x	xx.x
	SE	xx.x	xx.x	xx.x
Trt 1 (N=xxx)	n	xx	xx	xx
	LS Mean	xx.x	xx.x	xx.x
	SE	xx.x	xx.x	xx.x
	Comparison to Placebo			
	LS Mean Diff		xx.x	xx.x
	LS Diff SE		xx.x	xx.x
	P value		0.xxxx	0.xxxx
Trt 2 (N=xxx)	n	xx	xx	xx
	LS Mean	xx.x	xx.x	xx.x
	SE	xx.x	xx.x	xx.x
	Comparison to Reference			
	LS Mean Diff		xx.x	xx.x
	LS Diff SE		xx.x	xx.x
	P value		0.xxxx	0.xxxx

Figure 3. Table Layout #3

Many pharmaceutical companies already employ standard macros to conduct such analyses. These macros generate a specific table layout that can be selected and rerun if needed. However, in some cases, the requested layout might not be available, necessitating manual table generation. In these instances, the process can become time-consuming, particularly if the requested layout is complex or requires creating a new layout from scratch.

PROBLEM DECONSTRUCTION

To comprehensively address the issue, the problem was deconstructed into its constituent elements by examining three distinct table layouts:

- First, it was observed that all three tables contain the same fundamental information, rendering the repetition of the analysis unnecessary, as no new data would be generated in the process.
- Second, it was noted that the tables only differ in their presentation. The bottom-left table displays visits and statistics arranged by row, with treatment by column. In contrast, the top table displays statistics by column and visits and treatment by row. Finally, the third table arranges visits by column.
- Third, it is crucial to consider that, while not depicted in the examples, certain values may require reformatting, or specific statistics may become obsolete, leading to their removal from the display. This consideration is vital when analyzing data in real-world contexts.

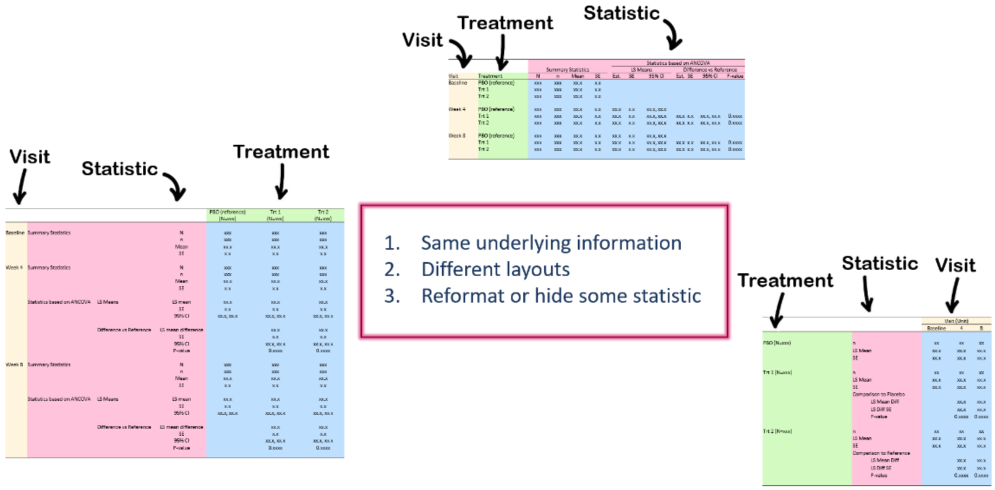

Figure 4. Problem Deconstruction

PAPER OUTLINE

This paper introduces an innovative solution to address the current challenges faced in table generation. Our team has developed a new display layout specification that will transform the way data is visualized. Additionally, we have created a flexible table generator macro that seamlessly supports the new layout specification. With this groundbreaking solution, data can be presented in a flexible and comprehensible manner, simplifying the process for statistical programmers to generate new layouts that meet their requirements.

The paper is organized as follows:

- We will first review the upstream process, including the preparation of data and the Analysis Results Datasets (ARD).
- Subsequently, we will explore table flexibility, covering both content and layout options.
- Lastly, we will discuss display layout specifications and explain how to generate various layouts using these specifications.

UPSTREAM PROCESS

A fundamental aspect of our proposal is the Analysis Results Standard (ARS) developed by CDISC. This standard streamlines analysis results by consolidating all results into a uniform format, enabling the generation and reuse of dynamic data displays. It should be noted that our company has been storing the results dataset prior to CDISC's development of ARD. Therefore, while the concept is consistent, the exact structure may differ. Despite the fact that ARS is still under development and its details may change in the future, we will be able to adapt effectively to the new standard.

The standard allows for consistent dissemination of data, facilitating more efficient communication and collaboration. With standardized analysis results, we can ensure that data is interpreted accurately and consistently across different teams and organizations. This is especially important in fields like clinical research, where data accuracy and reliability can have significant implications.

In summary, the ARS is a critical foundation for our proposal, and its importance should not be overlooked. It provides a standardized format for analysis results, enabling more efficient communication and collaboration, as well as ensuring data accuracy and reliability.

The following presents an exemplar dataset derived from our analytical process. The variables encompassed within are clearly delineated: a treatment variable (`ARM`/`ARMN`), visit variables (`AVISIT`/`AVISITN`), and statistic variables (`STATX`/`STATCD`). The column designated as `STATCD` comprises the statistic code, which serves an internal purpose within our organization. Each combination of these variables corresponds to a specific data value stored in the `RESULT` column. Additional columns, namely `STATCLS1` and `STATCLS2`, have been incorporated to facilitate the generation of nested column headers.

ARM	STATCLS1	STATX	STATCD	AVISIT	AVISITN	RESULT
PBO	Summary Statistics	N	BIGN	Baseline	0	xx
PBO	Summary Statistics	Mean	MEAN	Baseline	0	xx.x
PBO	Summary Statistics	SE	SE	Baseline	0	xx.x
PBO	Summary Statistics	n	N	Week 4	4	xx
PBO	Summary Statistics	Mean	MEAN	Week 4	4	xx
PBO	Summary Statistics	SE	SE	Week 4	4	xx.x
PBO	Statistics Based on ANCOVA	LS Mean	LSMEAN	Week 4	4	xx.x
PBO	Statistics Based on ANCOVA	SE	LSMEANSE	Week 4	4	xx
……	……	……	……	……	……	……
Trt 1	Summary Statistics	n	N	Week 4	4	xx
Trt 1	Summary Statistics	Mean	MEAN	Week 4	4	xx.x
Trt 1	Summary Statistics	SE	SE	Week 4	4	xx
Trt 1	Statistics Based on ANCOVA	LS Mean	LSMEAN	Week 4	4	xx.x
Trt 1	Statistics Based on ANCOVA	SE	SE	Week 4	4	xx.x
Trt 1	Statistics Based on ANCOVA	LS Mean Diff	LSMEANDIFF	Week 4	4	xx.x
Trt 1	Statistics Based on ANCOVA	LS Diff SE	LSMEANDIFFSE	Week 4	4	xx
Trt 1	Statistics Based on ANCOVA	P-Value	LSMEANPVAL	Week 4	4	xx.x
……	……	……	……	……	……	……
Trt 2	Summary Statistics	n	N	Week 8	8	Xx
Trt 2	Summary Statistics	Mean	MEAN	Week 8	8	xx.x
Trt 2	Summary Statistics	SE	SE	Week 8	8	xx.x
Trt 2	Statistics Based on ANCOVA	LS Mean Diff	LSMEANDIFF	Week 8	4	xx.x
Trt 2	Statistics Based on ANCOVA	LS Diff SE	LSMEANDIFFSE	Week 8	4	xx
Trt 2	Statistics Based on ANCOVA	P-Value	LSMEANPVAL	Week 8	4	xx.x
……	……		……	……	……	……

Figure 5. Example of Analysis Results Dataset

TABLE FLEXIBILITY

When defining a table, there are two aspects that may impact it:

1. **Content:** Content refers to the statistical values incorporated in the table. In the example below, there are three tables with distinct sets of statistics. The first table contains only basic summary and ANCOVA statistics, while the second table encompasses additional mean, standard error, and LS-means statistics. The third table features confidence intervals. It is essential to have the ability to select the variables we wish to display in the table based on varying scenarios.

		Statistics based on ANCOVA		
		Difference vs Reference		
N	n	Est.	95% CI	P-value

Summary Statistics				Statistics based on ANCOVA				
				LS Means		Difference vs Reference		
N	n	Mean	SE	Est.	SE	Est.	95% CI	P-value

Summary Statistics				Statistics based on ANCOVA						
				LS Means			Difference vs Reference			
N	n	Mean	SE	Est.	SE	95% CI	Est.	SE	95% CI	P-value

Figure 6. Content Flexibility

2. **Layout:** Layout pertains to the organization of the variables in our example. In this instance, the variables are visit, treatment, and statistics. We can arrange these variables either in columns or rows. For more complex tables, other elements such as sorting order, labeling, and nested headers also hold significance.

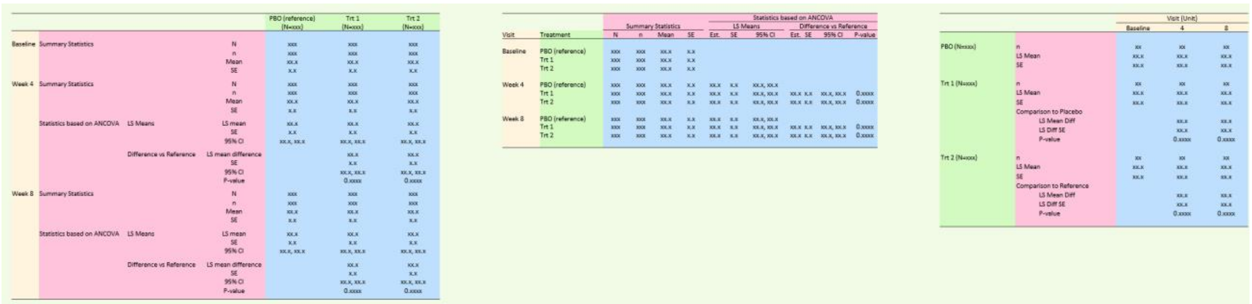

Figure 7. Layout Flexibility

With that in mind, we developed the Flexible Table Generator macro (`%flexible_table_generator`). We have incorporated content and layout flexibility into the macro.

```
%macro flexible_table_generator(
    ...
    statvar = /* List of statistics to display */,
    combvar = /* Combine variables into a new statistic to display */,
    ...
    colvar  = /* Variables used to define the columns */,
    rowvar  = /* Variables used to define the rows */,
    ...
);
```

Content is the first consideration. Utilizing <`STATVAR`> allows for the enumeration of all requisite statistics for display. The input order for these statistics holds significance, as it dictates their arrangement in rows or columns. Employing <`COMBVAR`> provides users with the capability to combine variables, thereby creating a novel statistic for presentation. Typical selections encompass counts and percentages, confidence intervals, among others.

For layout, users must define both <`COLVAR`> (column variables) and <`ROWVAR`> (row variables). These two inputs determine the basic table layout. There are sub-options for these two parameters, which we will discuss later. The macro also includes additional parameters such as <`DATAIN`> (input dataset). To simplify and focus, we will limit displayed parameters to those contributing to table flexibility.

CONTENT FLEXIBILITY

STATVAR Parameter

In our example, which involves analyzing results datasets, `STATX` represents the statistic header or description, and `STATCD` corresponds to the statistic code. In order to obtain the desired results, it is necessary to input the selected statistic code into the `STATVAR` parameter.

STATX	STATCD
N	BIGN
n	N
Mean	MEAN
SE	SE
LS Mean	LSMEAN
SE	LSMEANSE
LS Mean Lower Confidence Limit	LSMEANLCL
LS Mean Upper Confidence Limit	LSMEANUCL
LS Mean Diff	LSMEANDIFF
LS Diff SE	LSMEANDIFFSE
LS Diff Lower Confidence Limit	LSDIFFLCL
LS Diff Upper Confidence Limit	LSDIFFUCL
P-Value	LSMEANPVAL
…	

Figure 8. Input Selected Statistic Code

```
%macro flexible_table_generator(
```

```
    ...
    statvar = BIGN N LSMEANDIFF LSDIFFCL LSMEANPVAL,
    ...
);

%macro flexible_table_generator(
    ...
    statvar = BIGN N MEAN SE LSMEAN LSMEANSE LSMEANDIFF
                LSDIFFCL LSMEANPVAL,
    ...
);

%macro flexible_table_generator(
    ...
    statvar = BIGN N MEAN SE LSMEAN LSMEANSE LSMEANCL LSMEANDIFF
                LSMEANDIFFSE LSDIFFCL LSMEANPVAL,
    ...
);
```

All of these statistics originate from analysis result datasets. The macro is instructed to display these statistics in a table. It is important to note that the order of input variables directly impacts the table's layout. The display order of statistical variables is solely dependent on the order in which they are inputted during the macro call.

COMBVAR Parameter

For those interested in understanding the creation of the LS-mean difference confidence interval (LSDIFFCL) in previous section, it is important to note that it is derived from two distinct statistics: the LS-mean difference upper confidence limit (LSDIFFUCL) and lower confidence limit (LSDIFFLCL). To combine these variables and adjust their format, a method employing the <COMBVAR> parameter is provided. The process consists of two steps: firstly, specifying the type of combined variables as confidence interval ((alternative options encompass counts and percentages, minimum to maximum, Q1 to Q3, etc.); and secondly, inputting the included variables, in this case, the upper and lower confidence limits. Additionally, users can assign labels to the newly created variables.

```
%macro flexible_table_generator(
    ...
    combvar = LSDIFFCL(type=CI, vars=LSDIFFLCL LSDIFFUCL, label="95% CI"),
    ...
);
```

The diagram below demonstrates the data transformation process. The left side represents the storage of data in the analysis results dataset, while the right side displays the data in the table format.

STATX	STATCD	RESULT
LS Diff Lower Confidence Limit	LSDIFFLCL	82
LS Diff Upper Confidence Limit	LSDIFFUCL	94

→

STATX	STATCD	RESULT
95% CI	LSDIFFCL	(82, 94)

Figure 9. Combine Different Variables and Change Their Formats

LAYOUT FLEXIBILITY

In this section, a more thorough exploration of the layout aspect will be conducted. A conventional table layout encompasses three principal constituents: row headers, column headers, and a table body. Additional components, such as titles and footnotes, may also be incorporated.

- **Row/Column Headers:** These labels serve the purpose of identifying the data values present within the table. They may also encompass nested headers, which furnish more intricate details about the data.

- **Table Body:** This component houses all the significant data values. The headers provide a description of the data values in the table.

It is imperative to note that certain essential elements, such as the title and footnotes, have been excluded in order to preserve simplicity. Nonetheless, users retain the ability to modify these elements as required, utilizing the parameters provided.

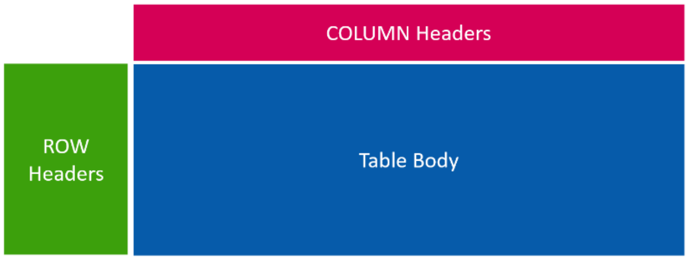

Figure 10. Table Components

To reconstruct the previously mentioned table layout, it can be dissected into three distinct parts, as previously delineated. The first table designates "Treatment" as the column variable and "Visit" and "Statistics" as the row variables. In the second table, the column variable transitions to "Statistics" and the row variables become "Visit" and "Treatment". Finally, in the third table, "Visit" is assigned as the column variable and "Treatment" and "Statistics" as the row variables. This table design model offers flexibility and allows us to view the data from different perspectives while remaining organized.

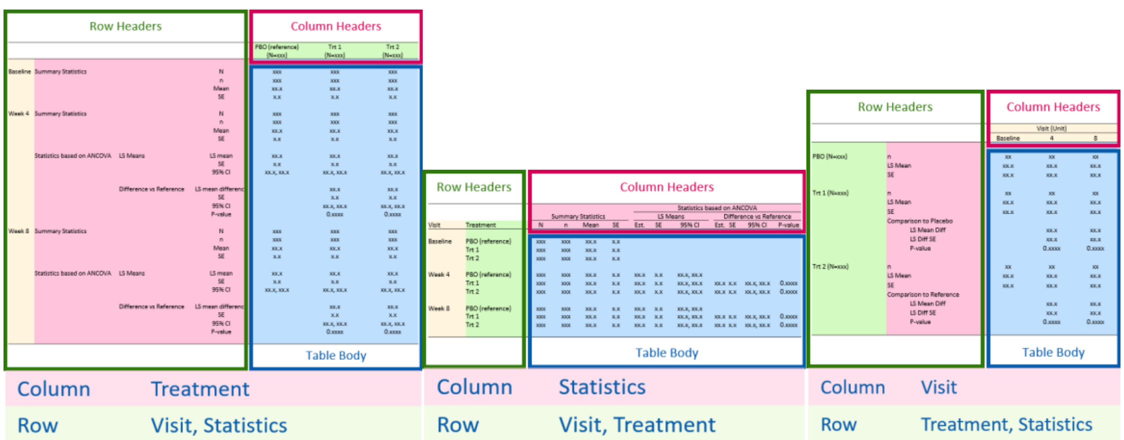

Figure 11. Layout Breakdown

Table Layout #1

To exemplify diverse table layouts, the identical analysis results dataset can be utilized as input for difference macro calls. The analysis results dataset is preserved, and different row and column variables are employed to regulate the table structure.

In order to generate the first table layout utilizing the analysis results dataset, the subsequent input parameters can be implemented:

- In the initial part, the <COLVAR> represents the Treatment. Within the macro, ARM is utilized as the <COLVAR> and the sort= option is employed to indicate the sorting variable, ARMN.
- The second part is organized by visit number and labeled "Visit", which serves as the table header.
- The third part is more intricate. As previously noted, the variables STATCLS1 and STATCLS2 have

been incorporated, functioning as row headers. These headers are displayed in separate columns and denote distinct categories of the statistics. In this table, two disparate header information instances are observed, both sorted by the `STATN` statistics number. This arrangement is contingent upon the user's input of the <`STATVAR`> parameter.

```
%macro flexible_table_generator(
    ...
    colvar = ARM (sort=ARMN),
    rowvar = AVISIT   (sort=AVISITN, label="Visit")
             STATCLS1 (sort=STATN)
             STATCLS2 (sort=STATN)
             STATX    (sort=STATN),
    ...
);
```

Table Layout #2

The second layout bears considerable resemblance to the first. Initially, the table incorporates statistics with nested headers at the top. To define the nested header, we utilize the `head=` option, which establishes different levels of headers for all the statistics. As we arrange the statistics based on user input order in the <`STATVAR`> section, a sorting variable is not necessary here. In the row variables, we have "Visit", sorted by visit number (`AVISITN`), and "Treatment", sorted by treatment number (`ARMN`).

```
%macro flexible_table_generator(
    ...
    colvar = STATX (head=STATCLS1 STATCLS2),
    rowvar = AVISIT (sort=AVISITN, label="Visit")
             ARM    (sort=ARMN, label="Treatment"),
    ...
);
```

This table layout presents the data differently from the first layout, with statistics serving as the column headers and visit and treatment as the row headers. By employing the macro with the appropriate input parameters, we can effortlessly generate table layouts that cater to various needs and preferences, enabling the presentation of data in a more customizable and adaptable manner.

Table Layout #3

In the third layout, we position the "Visit" variable in the columns and sort by visit number (`AVISITN`). We can also incorporate a header spanning across the columns using the `label=` option. For row variables, we arrange the "Treatment" variable first, followed by the statistics. We display the statistics somewhat differently from the first layout. Although there are two separate columns, we group them together and introduce indentations using the `indent=` option for all related variables.

```
%macro flexible_table_generator(
    ...
    colvar = AVISIT   (sort=AVISITN, label="Visit"),
    rowvar = ARM      (sort=ARMN,  label=)
             STATCLS1 (sort=STATN, label=, indent=Y)
             STATX    (sort=STATN, label=, indent=Y),
    ...
);
```

In this particular scenario, we successfully converted all three designs using the same set of analysis results. However, it is essential to recognize that numerous additional layouts may be available for use. Fortunately, we can still employ the aforementioned approach to effectively handle all these variations with ease and efficiency, ensuring the timely and accurate delivery of the desired results. The Flexible

Table Generator macro enables adaptation to different requirements and preferences, making it an indispensable tool for statistical programmers working with varying table layouts.

Sub-options

The following section enumerates the sub-options available for managing layouts. These sub-options have been developed to facilitate the customization of layouts according to specific requirements.

Option	Scope	Description
SORT	ROWVAR	Variable used to define the sort order
LABEL	ROWVAR, COLVAR	Define the row or column label to be used for the variable
INDENT	ROWVAR	Indent row header
HEAD	COLVAR	Define column header nesting

Figure 12. Sub-options

- **`sort=` option: This parameter determines the sorting order.** It is crucial to ensure proper sorting of tables. For instance, visit variables were sorted by `AVISITN` and treatments by `ARMN`, as specified in the macro call as sorting variables. Employing alphanumeric sorting may result in week 12 being sorted prior to week 4, an undesired outcome.

ARM	ARMN
PBO	0
Trt 1	1
Trt 2	2

AVISIT	AVISITN
Baseline	0
Week 4	1
Week 8	2
Week 12	3

Figure 13. Sorting Variables

- **`label=` option: Utilize the `label=` option to designate the row or column label for the variable.** This label functions as the table header. In the absence of a defined label option, the variable label serves as the header. However, when the label option is specified, the designated label is employed as the row header label or a spanning column header.

- **`indent=` option: This parameter enables row header indentation.** The `indent=` option permits the collapsing of individual column-based displays into a single column display featuring indentation. For instance, in the first example provided, variables under "Comparison to Placebo" are indented, whereas in the second example, each variable is presented as a separate column.

```
rowvar   =  statcls1(sort=statn, label=, indent=Y)
            statx    (sort=statn, label=, indent=Y)
```

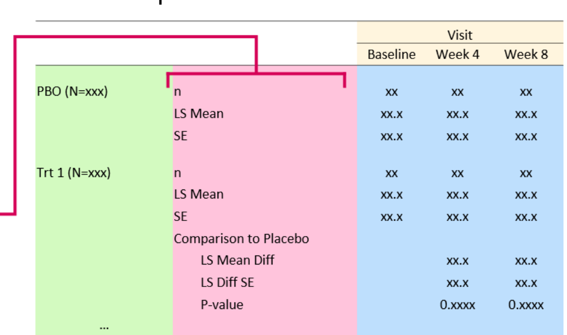

Figure 14. Single Column (Indented)

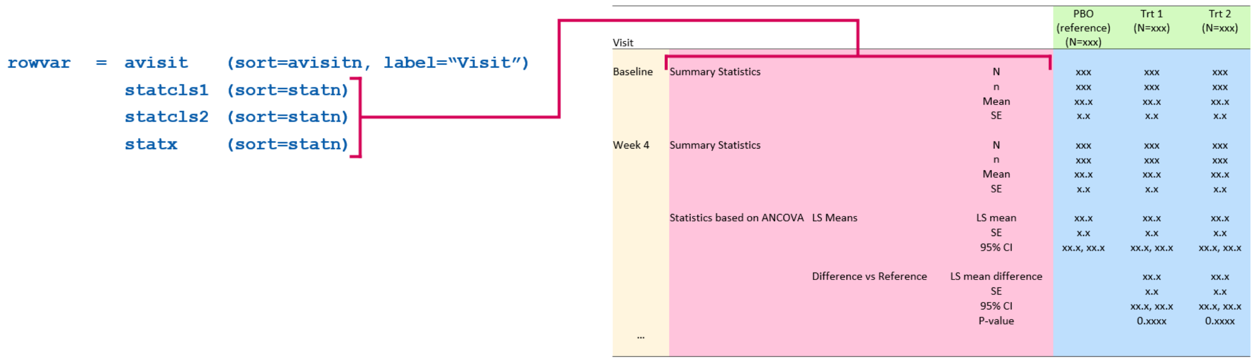

Figure 15. Multi-Columns

- **`head=` option: Define column header nesting.** To illustrate the creation of nested column headers, the `head=` option can be utilized within the macro to designate variables that function as headers. In the example provided, two column header variables, `STATCLS1` and `STATCLS2`, describe distinct categories of statistics, such as basic "Summary Statistics" and "Statistics based on ANCOVA". As shown in the figure, these two variables contribute to the construction of the nested column header arrangements. `STATX` represents the most fundamental level of header for the data values.

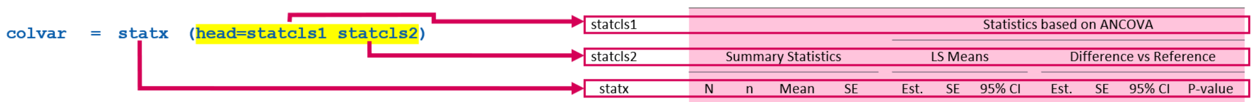

Figure 16. Nested Column Header Structure

Incorporating these sub-options accommodates a variety of table layouts, affording users the flexibility to adjust the table layout according to their distinct necessities.

DISPLAY LAYOUT SPECIFICATIONS

Display Layout Specifications provide an effective method for generating adaptable and customizable tables tailored to the requirements of diverse stakeholders. Employing machine-interpretable metadata to define the table guarantees that both the content and layout are pertinent and easily comprehensible for the target audience.

When utilizing Display Layout Specifications and a flexible table generator macro, the following key points should be taken into consideration:

1. Both content and layout are crucial components in table design. Thoughtful selection of statistical variables and their organization in a meaningful manner can facilitate valuable insights from the data.
2. A table generator macro enables flexibility in table design. This macro permits users to designate statistical variables, column and row variables, sorting order, nested headers, and additional elements, customizing the table to meet specific project needs.
3. Metadata can be employed to define and construct table designs. By entering parameters such as statistical variables (<`STATVAR`>), row variables (<`ROWVAR`>), and column variables (<`COLVAR`>), along with their associated sub-options, this information can be stored as metadata for future utilization.
4. This also facilitates the production of an increased number of pre-defined layouts utilizing the aforementioned metadata, rendering batch processing viable and streamlining standard analysis macros, as concerns regarding future table output are obviated.

In summary, the integration of Display Layout Specifications and a flexible table generator macro provides an efficacious approach for generating customizable and adaptable tables tailored to diverse project requirements. By taking into account both content and layout, as well as employing metadata to establish table designs, the presentation of data can be ensured to be both meaningful and comprehensible for an extensive array of stakeholders.

CONCLUSION

This paper examines the potential challenges encountered when generating diverse table layouts. Although table layouts exhibit considerable flexibility, encompassing all layout options within a standard macro design can prove arduous. To tackle this issue, the employment of the Analysis Results Dataset as an upstream process is recommended. This approach ensures the consistent formatting of all results, facilitating the development of a general macro capable of transforming and producing the desired table layout. The Display Layout Specifications and the flexible table generator macro serve as invaluable tools for defining and generating table outputs. By leveraging these tools, a dynamic and adaptable table layout can be readily achieved.

CONTACT INFORMATION

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

Songgu Xie
Regeneron Pharmaceutical, Inc.
(914) 479-2866
songgu.xie2@regeneron.com