PharmaSUG 2026 - Paper AP-116

# ARMed AutoTable Macro Agents: An ARM-Driven Framework for Automated Analysis Table Generation

Chengxin Li, AutoCheng Clinical Data Services LLC

## ABSTRACT

This paper presents AutoTable, a metadata-driven automation framework implemented entirely in SAS macros, where macro parameters are directly aligned with Analysis Results Metadata (ARM) framework. The system focuses on efficient, reproducible generation of non-inferential analysis tables while integrating seamlessly into existing statistical programming workflows.

Four core macro agents—%ADSLEVL, %BDSSTAT, %BDSSHIFT, and %OCCFREQ—support major categories of non-inferential analyses. For each macro invocation, the framework automatically generates:

- Formatted table output (RTF, PDF, or XLSX)
- Plain, executable, submission-ready SAS code
- Final results dataset (RDS)
- Aggregated ARM dataset (ARMDS) to facilitate define-xml generation
- Execution log file

By leveraging ARM as both specification and execution driver, AutoTable captures core structural features and layout patterns of table shells without requiring table shell standardization or digitization as prerequisites. The framework significantly improves programming efficiency compared to traditional copy-paste-update workflows while maintaining full transparency and traceability.

## INTRODUCTION

Automation of tables, listings, and figures (TLFs) has long been a primary objective in statistical programming. Despite advances in reusable code libraries and templated workflows, most clinical reporting environments still rely heavily on manual program modification. Copying legacy programs, adjusting logic incrementally, and revalidating outputs remain common practices. This approach is time-consuming, difficult to scale, and susceptible to inconsistency.

To address these limitations, the AutoTable framework—hereafter referred to as macro agents—was developed as a metadata-driven automation system. The design centers on Analysis Results Metadata (ARM), which is used not only for documentation but also as a structured configuration mechanism for SAS macros. Each macro call serves as a transparent, metadata-aligned execution unit capable of producing submission-ready outputs.

The primary design objectives are:

1. Cover approximately 95% of analytical tables and 85% of output layouts in clinical studies.
2. Improve efficiency compared to legacy copy-paste-update programming models.
3. Dynamically resolve plain executable SAS code that reproduces identical outputs.
4. Eliminate the prerequisites for table shell standardization or digitization.

This paper focuses on descriptive macro agents. Inferential macro agents are currently under development.

## ARM AS AN EXECUTION FRAMEWORK

Analysis Results Metadata (ARM) serves as structured table specifications, defining key attributes of each analysis result and ensuring traceability between outputs and their underlying ADaM datasets. While ARM is often used for documentation or define-xml support, AutoTable extends ARM to function as a direct execution framework.

Figure 1 illustrates the alignment between statistical programming processes and ARM.

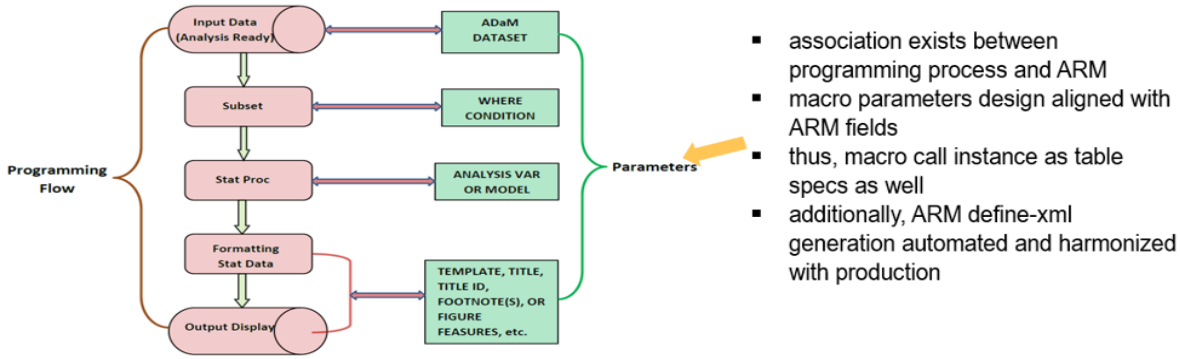


**Figure 1. Statistical Programming process Aligned with ARM**

Besides fully supporting SAP and traceability, analysis-ready is another core principle of ADaM development. Achieving autoTLF requires ADaM datasets to be analysis-ready, or pre-processed into an analysis-ready structure. After applying data-selection criteria (WHERE statement), the system can directly invoke SAS statistical procedures, formatting, and reporting.

Each macro call uses parameter values aligned with ARM fields, ensuring full consistency between metadata, specifications, and outputs.

## IMPLEMENTATION

AutoTable is currently implemented primarily in SAS, reflecting its continued dominance in regulatory clinical environments. The development roadmap includes progressive extension toward R-based implementations to support cross-language expansion.

The macro agents support flexible configuration at both table-level and parameter-level. Options include page breaks, sorting rules, indentation control, decimal precision, denominator definitions, and header transposition. Rather than enforcing rigid shell structures, the macros capture core structural patterns, thereby allowing compatibility with diverse table shell designs.

Three primary ADaM dataset structures support different analysis categories: ADSL, BDS, and OCCDS. Among these, shift tables derived from BDS datasets often present the highest degree of layout multiplicity. Correspondingly, four core macros have been developed covering non-inferential analyses as described below table 1:

| SAS Macro | Input ADaM | Analyses |
|---|---|---|
| %OCCFREQ | Any OCCDS, e.g., ADAE, ADCM, ADMH | Any frequency analyses |
| %BDSSTAT | Any BDS, e.g., ADLB, ADEG, ADVS | Any descriptive analyses with |

| | | |
|---|---|---|
| | | continuous/character variables |
| %BDSSHIFT | Any BDS, e.g., ADLB, ADEG | Shift table analyses, specifically |
| %ADSLEVL | ADSL | Subject evaluation analyses, e.g., tables of baseline, demographic, disposition, population, etc. |

**Table 1. AutoTable Scopes for Non-inferential Analysis**

Table 2 outlines the outputs from each macro call automatically.

| Seq | Item | Note |
|---|---|---|
| **1** | **Table** | table output in formats of RTF, PDF, or XLSX |
| **2** | **SAS Code** | Plain executable submission-ready SAS code dynamically resolved from the macro call. Executing the generated code reproduces the identical output as produced by the macro call |
| **3** | **RDS** | Final results dataset used for qc |
| **4** | **ARMDS** | Aggregated dataset of table specifications with ARM structure<br>Facilitate ARM define-xml generation |
| **5** | **Log file** | Macro execution logs |

**Table 2. AutoTable Output**

With %BDSSTAT or %ADSLEVL, principally numeric analysis variable triggers descriptive statistics (n, mean, median, sd, …), while char type variable triggers frequency summary (n, %). For descriptive statistics, decimal control can be achieved either through user specification or automatic detection from the input dataset.

One example of design of frequency analysis with OCCDS is depicted in figure 2.

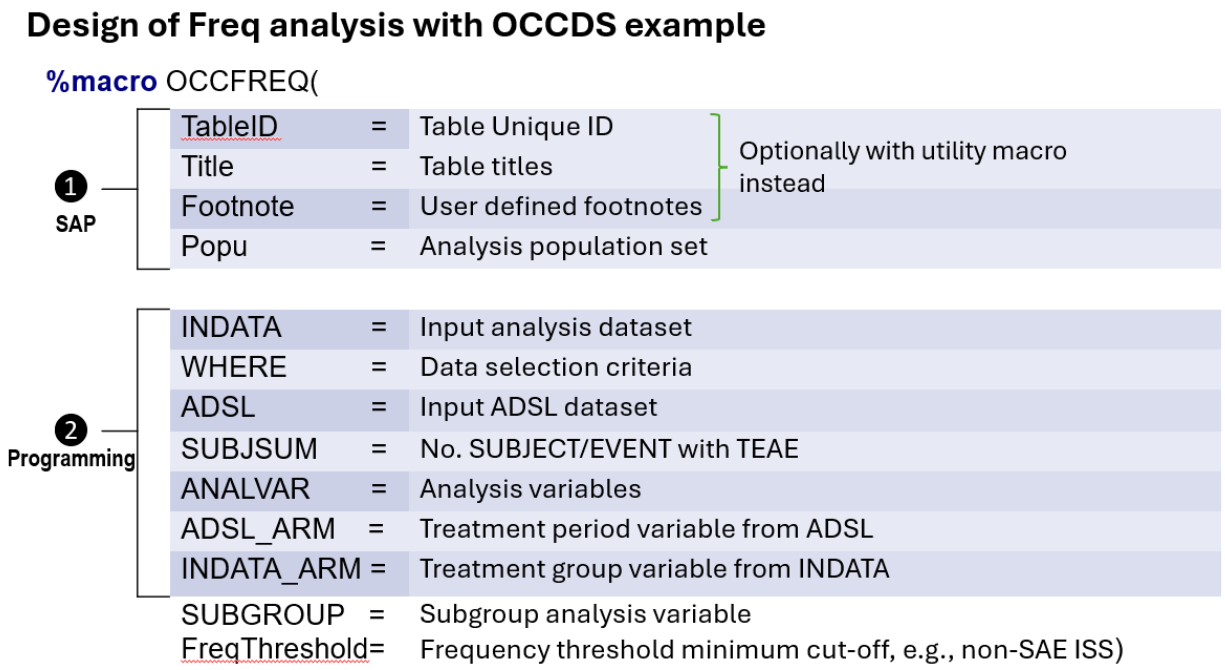


**Figure 2. Design of %OCCFREQ with ADaM.OCCDS**

## USE CASES

To illustrate the design and implementations of macro agents, representative examples of macro calls and resulting outputs are provided. The simulated ADaM datasets (ADSL, ADLB, ADVS, and ADAE) are further developed based on the templates in RDS v1.0 of CDISC.ORG. For any macro call instance, the corresponding plain executable and submission-ready SAS code in a txt file is resolved. The identical table will be produced if execute the code (omitted here).

### 1. DEMOGRAPHIC TABLE WITH %ADSLEVL

This example demonstrates how %ADSLEVL reads ARM-aligned parameters to produce demographic table.

```
%ADSLEVL(
      TABLEID=%NRBQUOTE(t_demo2\LCHLBL=Category), /*TableID required; LCHLBL option to update the default display label*/
      TITLE=,    /*opt., separated with # if multiple. if both TITLE and FOOTNOTE NULL indicating user with utility macro*/
      FOOTNOTE=, /*opt., separated with # if multiple*/

      POPU=SAFFL, /*POPU parameter for big N summary as well*/

      ADSL=%str(ADaM.ADSL),
      WHERE =%str(SAFFL='Y'),   /*data selection criteria from ADSL, applied to all analysis variables lised in ANALVAR parameters*/

      ADSL_ARM=%NRBQUOTE(TRT01A\PRELOADFMT=$TRTA.\INFORMAT=TRTAN.\THCLIST=Xanomeline|[TRT11 TRT12 TRT19] ~ TRT21 ~ TRT99\THCFMT=$TRTSHEADER.),
                         /*PRELOADFMT option for trt grouping, THCLIST option for spanning header, THCFMT option for further fomatting*/

      ANALVAR=%NRBQUOTE(AGEGR1\CATSUM=OLI\DISP=%nrbquote(Age (years), n (%)) # /*CATSUM option for variable layout pattern*/
                        AGE\DISP=Age (years) # /*DISP option for variable label display, by default, with variable label*/
                        ARACE\CATSUM=OLI\preloadfmt=$RACE.\informat=RACEN.\DISP=%nrbquote(Race, n (%)) #
                        AETHNIC\CATSUM=OLI\DISP=%nrbquote(Ethnicity, n (%)) #
                        WEIGHTBL # HEIGHTBL # BMIBL # ACOUNTRY\DISP=Country\informat=COUNTRYN. /*informat option for sort*/
                        )  /*CHAR var triggering freq summary, NUM var triggering DSTAT summary*/
   );
```

**Macro Call Instance 1. Demographic Table with %ADSLEVL**

| Category | XANOMELINE | | | Placebo | All Subjects |
|---|---|---|---|---|---|
| | Low Dose (N=84) | High Dose (N=84) | Pooled (N=168) | (N=86) | (N=254) |
| Age (years), n (%) | | | | | |
| <65 | 8 ( 9.5) | 11 (13.1) | 19 (11.3) | 14 (16.3) | 33 (13.0) |
| 65-80 | 47 (56.0) | 55 (65.5) | 102 (60.7) | 42 (48.8) | 144 (56.7) |
| >80 | 29 (34.5) | 18 (21.4) | 47 (28.0) | 30 (34.9) | 77 (30.3) |
| Age (years) | | | | | |
| n | 84 | 84 | 168 | 86 | 254 |
| Mean (SD) | 75.7 (8.29) | 74.4 (7.89) | 75.0 (8.09) | 75.2 (8.59) | 75.1 (8.25) |
| Median | 77.5 | 76.0 | 77.0 | 76.0 | 77.0 |
| Min, Max | 51, 88 | 56, 88 | 51, 88 | 52, 89 | 51, 89 |

**Macro Call Instance 1 Output. Demographic Table with %ADSLEVL**

## 2. OBSERVED VALUES, CHG BY VISIT TABLE WITH %BDSSTAT

This example demonstrates how %BDSSTAT configured with ARM-aligned parameters to produce lab values and changed from baseline table.

```
%BDSSTAT(
    TABLEID=%str(t_lab_saf),

    POPU=SAFFL,  /*SAFFL is default population as well*/

    INDATA=ADaM.ADLB,
    where=%str(AVISITN>=0),
    ADSL=ADaM.ADSL,

    ANALVAR=%NRBQUOTE(BASE\disp=Baseline # AVAL # CHG),
    BYVAR=%str(PARAM\DPAT=TV # AVISIT\DPAT=LC), /*DPAT option making possible to support more flexible table shells*/

    ADSL_ARM=TRT01A,
    INDATA_ARM=%NRBQUOTE(TRTA\PRELOADFMT=$TRTA.\INFORMAT=TRTAN.)
    );
```

**Macro Call Instance 2. Lab Values and Changed from Baseline Table with %BDSSTAT**

Parameter = Hemoglobin (mmol/L)

| Analysis Visit Statistics | Xanomeline Low Dose (N=84) | Xanomeline High Dose (N=84) | Pooled Xanomeline (N=168) | Placebo (N=86) | All Subjects (N=254) |
|---|---|---|---|---|---|
| Baseline | | | | | |
| n | 81 | 81 | 162 | 85 | 247 |
| Mean (SD) | 8.62 (0.768) | 8.87 (0.783) | 8.75 (0.784) | 8.62 (0.828) | 8.70 (0.800) |
| Median | 8.56 | 8.87 | 8.75 | 8.63 | 8.69 |
| Min, Max | 6.6, 10.2 | 7.4, 10.6 | 6.6, 10.6 | 6.4, 10.4 | 6.4, 10.6 |
| Week 2 | | | | | |
| n | 80 | 80 | 160 | 83 | 243 |
| Mean (SD) | 8.45 (0.763) | 8.72 (0.829) | 8.58 (0.805) | 8.44 (0.765) | 8.53 (0.793) |
| Median | 8.44 | 8.66 | 8.53 | 8.50 | 8.50 |
| Min, Max | 6.5, 10.5 | 7.1, 10.9 | 6.5, 10.9 | 6.8, 10.4 | 6.5, 10.9 |
| Change from Baseline | | | | | |
| n | 77 | 77 | 154 | 82 | 236 |
| Mean (SD) | -0.16 (0.462) | -0.21 (0.425) | -0.18 (0.443) | -0.16 (0.416) | -0.17 (0.433) |
| Median | -0.06 | -0.25 | -0.19 | -0.19 | -0.19 |
| Min, Max | -1.2, 0.7 | -1.3, 1.2 | -1.3, 1.2 | -1.1, 0.9 | -1.3, 1.2 |

**Macro Call Instance 2 Output. Lab Values and Changed from Baseline Table with %BDSSTAT**

To align with the provided table shell requesting analysis variable transposed to the spanning table header, the tableID parameter can be configured with ADVARDPAT option:

```
TABLEID=%NRBQUOTE(t_lab_saf2\LCHLBL=%NRBQUOTE(Visit^n   Statistics)\AVARDPAT=SH\LCPCHG=1%),
/*analysis variables transposed to header display with option setting, left column width/label adjustable*/
```

The corresponding output:

Parameter = Hemoglobin (mmol/L)

| | Xanomeline Low Dose (N=84) | | Xanomeline High Dose (N=84) | | Pooled Xanomeline (N=168) | | Placebo (N=86) | | All Subjects (N=254) | |
|---|---|---|---|---|---|---|---|---|---|---|
| Visit Statistics | Observed Value | Change from Baseline | Observed Value | Change from Baseline | Observed Value | Change from Baseline | Observed Value | Change from Baseline | Observed Value | Change from Baseline |
| Baseline | | | | | | | | | | |
| n | 81 | | 81 | | 162 | | 85 | | 247 | |
| Mean (SD) | 8.62 (0.768) | | 8.87 (0.783) | | 8.75 (0.784) | | 8.62 (0.828) | | 8.70 (0.800) | |
| Median | 8.56 | | 8.87 | | 8.75 | | 8.63 | | 8.69 | |
| Min, Max | 6.6, 10.2 | | 7.4, 10.6 | | 6.6, 10.6 | | 6.4, 10.4 | | 6.4, 10.6 | |
| Week 2 | | | | | | | | | | |
| n | 80 | 77 | 80 | 77 | 160 | 154 | 83 | 82 | 243 | 236 |
| Mean (SD) | 8.45 (0.763) | -0.16 (0.462) | 8.72 (0.829) | -0.21 (0.425) | 8.58 (0.805) | -0.18 (0.443) | 8.44 (0.765) | -0.16 (0.416) | 8.53 (0.793) | -0.17 (0.433) |
| Median | 8.44 | -0.06 | 8.66 | -0.25 | 8.53 | -0.19 | 8.50 | -0.19 | 8.50 | -0.19 |
| Min, Max | 6.5, 10.5 | -1.2, 0.7 | 7.1, 10.9 | -1.3, 1.2 | 6.5, 10.9 | -1.3, 1.2 | 6.8, 10.4 | -1.1, 0.9 | 6.5, 10.9 | -1.3, 1.2 |
| Week 4 | | | | | | | | | | |
| n | 71 | 69 | 71 | 69 | 142 | 138 | 79 | 78 | 221 | 216 |
| Mean (SD) | 8.35 (0.780) | -0.30 (0.501) | 8.62 (0.780) | -0.21 (0.452) | 8.49 (0.789) | -0.25 (0.477) | 8.37 (0.815) | -0.23 (0.436) | 8.45 (0.798) | -0.25 (0.462) |
| Median | 8.19 | -0.25 | 8.56 | -0.25 | 8.38 | -0.25 | 8.32 | -0.31 | 8.38 | -0.25 |
| Min, Max | 6.1, 10.1 | -2.4, 0.9 | 7.4, 10.8 | -1.2, 1.0 | 6.1, 10.8 | -2.4, 1.0 | 6.5, 10.1 | -1.1, 1.1 | 6.1, 10.8 | -2.4, 1.1 |

**Macro Call Instance 3 Output. Lab Values and Changed from Baseline Table with %BDSSTAT**

With %BDSSTAT, it can support any frequency summaries with BYVAR enabled or disabled, denominator with N or m, just named a few, with AVALC, AVALCAT, CRIT/MCRIT, completeness or accumulated counting, etc.

## 3. SOC, PT, AND SEV TABLE WITH %OCCFREQ

Basically, with OCCDS, the summary often features the embedded layered counting, e.g., by AESOC and AEDECOD. For severity or toxicity grade related counting, counting the worst only is also required. For this type of layered counting, the macro agent defines the option(INDENT) to align with the table shell.

```
%OCCFREQ(
    TABLEID=%NRBQUOTE(t_teae_socptsev), /*left column header autogenerated by default, further customerized if needed*/
    Title  =%str(Table 14.3.1.2 Treatment-Emergent Adverse Events by System Organ Class, Preferred Term and Severity # (Safety Analysis Set)),
    Footnote=%NRBQUOTE(TEAE: treatment-emergent adverse event. #
                       Note: Adverse events (AEs) are coded using MedDRA version 27.0.),
                      /*commonly titles/footnotes loaded with company utility macro instead of manual input here*/
    Popu   =%str(SAFFL),
    Indata =ADaM.ADAE,     /*or WORK.ADAE if preprocessed*/
    ADSL   =ADaM.ADSL,
    SUBJSUM=,              /*%str(EVENT\DISP=No. of Events # USUBJ\DISP=Subjects with at least one TEAE)*/

    ANALVAR=%NRBQUOTE(ASEV\WHERE=AOCCIFL='Y'\preloadfmt=$ASEV.\informat=ASEVN.\CATSUM=OLI\disp=TEAE #
                      AESOC\DISP=System Organ Class\INDENT=0\SUMOFF=Y #   /*INDENT option defining embeded layer associations*/
                        ASEV\WHERE=AOCCSIFL='Y'\preloadfmt=$ASEV\informat=ASEVN.\INDENT=1 #
                        AEDECOD\DISP=Preferred Term\INDENT=1\SUMOFF=Y #
                          ASEV\WHERE=AOCCPIFL='Y'\preloadfmt=$ASEV.\informat=ASEVN.\INDENT=2
                      ),

    ADSL_ARM=%str(TRT01A),
    INDATA_ARM=%NRBQUOTE(TRTA\PRELOADFMT=$TRTA.\INFORMAT=TRTAN.\THCLIST="Xanomeline"|[TRT11 TRT12 TRT19] ~ TRT21 ~ TRT99\THCFMT=$TRTSHEADER.)
);  /*THCLIST for table spanning header, TCHFMT further formating header display if needed*/
```

**Macro Call Instance 4. SOC, PT, and Sev Table with %OCCFREQ**

Table 14.3.1.2 Treatment-Emergent Adverse Events by System Organ Class, Preferred Term and Severity
(Safety Analysis Set)

| System Organ Class<br>Preferred Term<br>Analysis Severity/Intensity | Xanomeline<br>Low Dose<br>(N=84) | Xanomeline<br>High Dose<br>(N=84) | Xanomeline<br>Pooled<br>(N=168) | Placebo<br>(N=86) | All Subjects<br>(N=254) |
|---|---|---|---|---|---|
| TEAE | | | | | |
| Mild | 0 | 2 ( 2.4) | 2 ( 1.2) | 3 ( 3.5) | 5 ( 2.0) |
| Moderate | 41 (48.8) | 37 (44.0) | 78 (46.4) | 41 (47.7) | 119 (46.9) |
| Severe | 36 (42.9) | 40 (47.6) | 76 (45.2) | 25 (29.1) | 101 (39.8) |
| GENERAL DISORDERS AND ADMINISTRATION SITE CONDITIONS | | | | | |
| Mild | 1 ( 1.2) | 0 | 1 ( 0.6) | 1 ( 1.2) | 2 ( 0.8) |
| Moderate | 39 (46.4) | 37 (44.0) | 76 (45.2) | 19 (22.1) | 95 (37.4) |
| Severe | 7 ( 8.3) | 3 ( 3.6) | 10 ( 6.0) | 1 ( 1.2) | 11 ( 4.3) |
| APPLICATION SITE PRURITUS | | | | | |
| Mild | 0 | 0 | 0 | 0 | 0 |
| Moderate | 18 (21.4) | 20 (23.8) | 38 (22.6) | 6 ( 7.0) | 44 (17.3) |
| Severe | 4 ( 4.8) | 2 ( 2.4) | 6 ( 3.6) | 0 | 6 ( 2.4) |

**Macro Call Instance 4 Output. SOC, PT, and Sev Table with %OCCFREQ**

In %OCCFREQ development, to support any frequency summary, the primary technical challenge is to generate REPORT-ready output using a universal sorting strategy applicable across diverse frequency structures.

## 4. SHIFT TABLE (FROM BASE TO POST-BASE) WITH %BDSSHIFT

Shift tables often exhibit substantial layout flexibility. With settings, the SHIFT table agent can easily implement:

1) Either BASELINE or POST-BASELINE transposed
2) Denominator with N, m, or n
3) Combinations of TRTA, subgroup, PARAM, AVISIT presented in title (with byVALn), left column, or spanning header
4) Multiple shifts appended in one table

Conforming with table shell, below call instance generates one table with treatment group and VISIT displayed at left column, PARAM at title.

```
/*with table shell requested, treatment group and visits displayed at left column; PARAM at title*/
%BDSSHIFT(
    TABLEID=t_lab_shift1,
    POPU=SAFFL,

    INDATA=%str(ADaM.ADLB),
    WHERE =%NRBQUOTE(PARAMCD in ('ALT', 'AST', 'CK')),

    ADSL=%str(ADaM.ADSL),

    ANALVAR=%NRBQUOTE(BNRIND\preloadfmt=$NRIND.\informat=NRINDN. ~ ANRIND\preloadfmt=$NRIND.\informat=NRINDN.),
    /*the 2nd paired variable(here ANRIND) always transposed. if BNRIDN transposed, then BNRIND listed to 2nd;
      multi-pairs supported with '#' separated, and further with WHERE sub-option enabled to select the data.*/

    BYVAR=%str(PARAMCD\DPAT=TV # AVISIT\DPAT=LC),

    ADSL_ARM=%str(TRT01A),
    INDATA_ARM=%str(TRTA\DPAT=LC)
);
```

**Macro Call Instance 5. Shift Table (from BASE to Post-BASE) with %BDSSHIFT**

The corresponding generated table with partial screen shot:

Parameter: Alanine Aminotransferase (U/L)

| Treatment Group | Baseline Category | Post-Baseline Category, n (%) | | | |
|---|---|---|---|---|---|
| | | Low | Normal | High | Total |
| Xanomeline Low Dose (N=84) | Week 2 | | | | |
| | Low | 2 ( 2.6) | 1 ( 1.3) | 0 | 3 ( 3.8) |
| | Normal | 0 | 73 (93.6) | 0 | 73 (93.6) |
| | High | 0 | 1 ( 1.3) | 1 ( 1.3) | 2 ( 2.6) |
| | Total | 2 ( 2.6) | 75 (96.2) | 1 ( 1.3) | 78 (100) |
| | Week 4 | | | | |
| | Low | 2 ( 2.9) | 0 | 0 | 2 ( 2.9) |
| | Normal | 2 ( 2.9) | 64 (91.4) | 0 | 66 (94.3) |
| | High | 0 | 1 ( 1.4) | 1 ( 1.4) | 2 ( 2.9) |
| | Total | 4 ( 5.7) | 65 (92.9) | 1 ( 1.4) | 70 (100) |

**Macro Call Instance 5 Output. Shift Table (from BASE to Post-BASE) with %BDSSHIFT**

## AI-POWERED AUTOPROGRAMMING

The rapid development of AI/ML technologies presents new opportunities for statistical programming. However, with explorations and investigations, significant validation and governance challenges remain for fully automated programming within regulated clinical study environments, although strong potential exists for documentation and configuration assistance, e.g., annotations, specifications, review guide drafting.

Under the current implementation model, one proposed architecture is illustrated in Figure 3.

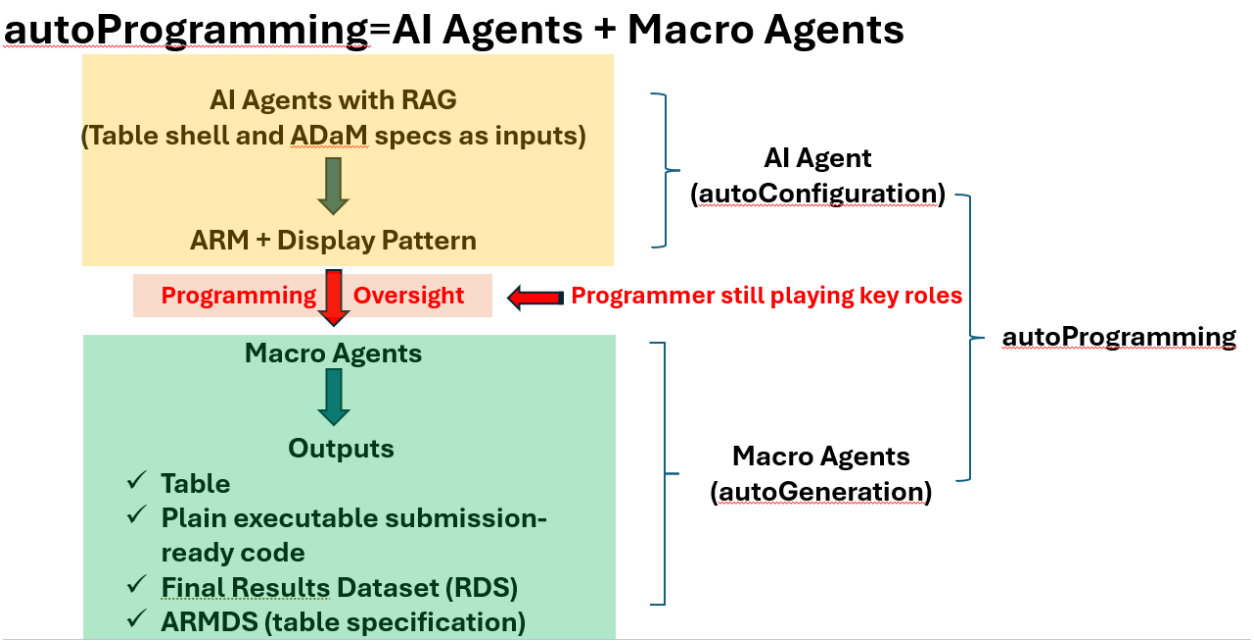


**Figure 3. AI Powered AutoProgramming**

This hybrid model preserves macro-based determinism and production reliability while leveraging AI for interpretive and configuration support. In this design, macro agents remain the deterministic execution engine, and AI serves as a controlled assistive layer.

## AUTOTLF ROADMAP

The AutoTable project is part of a broader **autoTLF** roadmap as shown in table 3 below.

| Stage | Scope | Status |
|---|---|---|
| **Non-Inferential Tables** | %ADSLEVL, %OCCFREQ, %BDSSTAT, %BDSSHIFT | Completed |
| **Inferential Tables** | %ANCOVA, %MMRM, %CMH, %TTE | In Progress |
| **AI/ML Powered Agents** | AI-based interpretation and automation | Initiated |
| **Figures and Listings** | Expansion beyond tables | Planned |
| **R/Python Packages** | Cross-language support | Under Evaluation |

**Table 3. AutoTLF Roadmap**

Future releases will extend automation to inferential analyses and further integrate AI agents for structured document interpretation and configuration support. Long-term plans include migration toward multi-language support (SAS, R, Python) and hybrid architectures combining AI-driven reasoning with macro-based agents.

## CONCLUSION

AutoTable demonstrates that Analysis Results Metadata (ARM) can function not only as documentation but also as an execution framework for automated table generation. By aligning SAS macro parameters directly with ARM fields, the macro agents transform table specifications into transparent, reproducible, and submission-ready outputs.

The design objective—to cover approximately 95% of analytical tables and 85% of output layouts in clinical studies—is intentionally conservative. Through structured parameterization and layout pattern capture, the macro agents substantially reduce the need for repetitive copy-paste-update programming practices. Instead of rewriting programs for each table, programmers configure specifications through macro parameters, allowing table generation to focus on structured metadata rather than procedural code replication.

Each macro invocation simultaneously produces formatted output (RTF/PDF/XLSX), results dataset (RDS), ARM-compliant metadata (ARMDS), and plain executable SAS code that reproduces identical results. This dual-generation approach enhances transparency, facilitates quality control, and supports define-xml integration while preserving compatibility with existing workflows.

Importantly, AutoTable macro agent does not require prior table-shell standardization or digitization. By capturing core structural patterns—such as header configurations, denominator logic, transposition options, and layout multiplicity—the framework supports diverse table shells without imposing rigid formatting constraints. At the same time, organizations that have standardized shells can further benefit from automation efficiencies.

Integration with AI/ML technologies represents a complementary extension rather than a replacement of the macro-agent architecture. While AI-powered autoProgramming introduces promising capabilities for configuration assistance and documentation support, the current design retains programmer oversight and preserves established production models within clinical statistical programming environments.

Overall, AutoTable macro agent provides a scalable, production-compatible framework that enhances efficiency, maintains traceability, and consolidates routine programming practices. By shifting focus from procedural scripting to metadata-driven specifications, the framework establishes a practical foundation for intelligent, controlled automation in clinical reporting.

As regulatory expectations for traceability and efficiency continue to increase, metadata-driven execution frameworks such as AutoTable macro agents provide a practical and scalable path forward.

## CONTACT INFORMATION

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

Chengxin Li
AutoCheng Clinical Data Services LLC
Chengxin.li@autoclindata.com