PharmaSUG 2025 - Paper ST-168

The Allowable Total Difference Zone: A construction method using the ATDzone SAS® Macro

Jesse A. Canchola, Roche Molecular Diagnostics, Inc., Pleasanton, California, USA
Natasha Oza, Roche Molecular Diagnostics, Inc., Pleasanton, California, USA

ABSTRACT

When comparing two systems using the same item or “sample” to produce at least two “paired” results (one on each system), for example, a new versus an older system for a molecular assay, typical method comparison methodologies used are Ordinary Least Squares (OLS), Deming, or Passing-Bablok regression (Passing & Bablok, 1983), bias plots that include Bland-Altman and Krouwer bias plot analysis and Error Grid Analysis (EGA) (Passing & Bablok, 1983; Linnet, 1993; Altman & Bland, 1983; Bland & Altman, 1986; Clark et al., 1987; Parkes et al., 2000). One additional enhancement to most of these methods, not typically used for this type of analysis, called Allowable Total Difference (ATD) zone (CLSI EP21-A, 2003; Krouwer & Monti, 1995; Krouwer, 2008), utilizes the reproducibility results of the older system (for example, from package/product inserts or from product requirements document) to construct the boundaries or limits that define where 95% of the differences between the two repeated measurements by the older system should be inside of those limits in the context of a Bland-Altman or Krouwer bias plot. One application of the ATD Zone include using it in a bias scatterplot of a new system (Y) compared to an older system (X). Producing ATD Zone plots can be a challenging programming endeavor. However, the authors a introduce a SAS® macro, *ATDzone*, that simplifies their creation for any method comparison task at hand, using the Bland-Altman bias plots, with minimal inputs.

INTRODUCTION AND BACKGROUND

Typical method comparison bias studies are analyzed using parametric methods such as ordinary least squares (OLS) regression (aka OLR or Ordinary Linear Regression), Deming regression or Passing-Bablok (PB) regression. Additionally, bias plots are used to evaluate the level of bias between the two systems [aka New or Test method (X) vs. Older or Reference method (Y)]. Let D = Y – X on the y-axis and, on the x-axis, one can have the X (Krouwer bias plot; Krouwer, 1987; Krouwer & Cembrowski, 1991) or the X and Y average (i.e., (X+Y)/2), the Bland-Altman bias plot (Bland & Altman, 1986). To have a fair comparison between the two systems, the same item or “sample” is used to generate paired data between each system.

These plots can additionally be enhanced with the Allowable Total Difference (ATD) zone information to visualize and quantify the percentage of data in the zone variability sectors of the method. Basically, the ATD Zone is constructed such that the reproducibility results of the older system are used to create the boundaries or limits that define where 95% of the differences between the two repeated measurements by the older system should be inside of those limits. We introduce the SAS® macro, *ATDzone*, that creates the zone for any method comparison task at hand with minimal inputs.

FDA Context for ATD Zone (Allowable Total Difference) and Cutoff Determinations

According to the U.S. Food and Drug Administration's *Assay Migration Studies for In Vitro Diagnostic Devices—Guidance for Industry and FDA Staff (2013)*, sponsors transferring an in vitro diagnostic assay from an "old" (previously approved/cleared/licensed) system to a "new" system are advised to demonstrate that "there are no changes to performance characteristics that could affect safety and effectiveness."

In particular, the FDA emphasizes defining acceptance criteria based on the old system's own measurement variability (e.g., from within-laboratory precision and reproducibility studies). By comparing new-system results to "the distribution of numeric values that the old system would itself generate if tested in replicate" [Section VI.A of the FDA Guidance], the sponsor can establish whether the new system's results remain "close enough" to those from the old system.

To implement this recommendation, many developers (including this paper's authors) use an *Allowable Total Difference (ATD)* zone. The ATD zone sets upper and lower boundaries around the old system's expected result, anchored by:

1. The systematic bias (if any) observed between the old and new systems in a method-comparison or regression analysis, and
2. The old system's reproducibility estimates, usually the standard deviation (SD) (or percent coefficient of variation, CV) derived from multi-day or multi-site precision studies.

By defining these boundaries so that approximately 95% of the old system's repeated measurements would lie within them (i.e., ±2 SD, or another chosen coverage range), the new system's reading for the same sample "should similarly fall within that same zone" if the performance has not changed meaningfully. This approach is consistent with guidance stipulating that "the systematic difference between numeric values of the new and old systems should be either not clinically significant and not statistically significant or remain within established acceptance criteria based on old-system reproducibility" [Section VI.A.4].

Furthermore, for borderline or equivocal samples, the guidance recommends that sponsors "present an allowable total difference zone and demonstrate that the new-system measurements remain inside this zone for the majority of samples." This ensures that minor fluctuations near the cutoff or within a re-test zone do not reflect a clinically problematic shift.

In summary, we derive our ATD zone boundaries from the old system's published reproducibility profile (e.g., from the package insert or in-house data) and then incorporate them into our method-comparison plots. For each sample, we assess whether the new system's result lies within those ± limits around the old system's value. In so doing, we follow the FDA-recommended strategy of relying on "the natural variability between two measurements by the old system" as the benchmark for allowable difference [FDA 2013, Sections VI.A-B].

2. Methodology

2.1 ATD Zone Definition

We defined the ATD zone in alignment with the FDA's *Assay Migration Studies for In Vitro Diagnostic Devices* recommendations for allowable difference. Specifically, we utilized the old system's standard deviations (SD) at relevant analyte/method levels—obtained from its existing reproducibility data—to establish ±2•SD boundaries around a regression-predicted old-system value. We additionally incorporated any known systematic bias when centering these boundaries. Following the FDA's recommendation to 'demonstrate that the new system's measurement remains within the old system's variation profile,' any new-system result lying outside these boundaries was deemed potentially discrepant and worthy of further scrutiny.

To reiterate, the ATD Zone is determined using reproducibility results from the old system, setting boundaries where 95% of measurement differences between the new and old systems should fall. The boundaries are derived using:

- Systematic or proportional bias estimates from Bland-Altman analysis
- Standard deviations from reproducibility studies

Specifically, the formulas used are as follows.

ATD zone construction Steps:

a) Use the standard deviation (SD or σ) from the old system's reproducibility at certain method (or analyte) levels.

b) (optional) Adjust σ to $\sigma_{adj} = \sigma \cdot f$ (1)

where $f = \left[\left(1 - \frac{1}{4\nu}\right)^{-1} \cdot \sqrt{\frac{\nu}{\chi^2_{\alpha,\nu}}}\right]$, and where $\chi^2_{\alpha,\nu}$ is the lower quantile of the chi-square distribution with $\nu\ degrees\ of\ freedom$ at a chosen significance level alpha (α).

c) Construct the bounding lines in a Bland-Altman bias scatterplot or define the numeric threshold for difference:

$$Lower = \left(X_{old} + \hat{b}\right) - z \cdot \sqrt{2} \cdot \sigma_{adj} \quad (2)$$

$$Upper = \left(X_{old} + \hat{b}\right) + z \cdot \sqrt{2} \cdot \sigma_{adj} \quad (3)$$

where $\hat{b}$ is an optional added known bias of the new relative to the old method (currently set to zero in the macro) and where $z \approx 1.96$ for 95% coverage.

These steps mirror the approach used in our ATDzone SAS macro, as well as the conceptual instructions in the FDA assay migration guidance and CLSI EP21 for total analytical error.

2.2 Statistical Analysis

Key statistical methods utilized include:

- **Bland-Altman Plots**: To model measurement difference relationships and estimate systematic bias.
- **Precision Estimates**: To assess within-laboratory and between-site variability.

3. SAS Macro Implementation

3.1 Macro Overview

The latest ATDzone SAS macro, downloadable online (http://bit.ly/4hHpKfq) or in the Appendix below, automates:

- Data preprocessing
- Calculation of ATD boundaries and shading
- Regression modeling
- Visualizations and summary reporting

3.2 Macro Code Structure and Inputs

HbA1c Example

To help understand the different macro code structure and inputs, let us begin with an example using hemoglobin A1c. Now, hemoglobin A1c is often abbreviated as HbA1c or is simply known as A1c (going forward in this discussion), is a blood test that measures your average blood sugar levels over the past two to three months. Here's how it works (American Diabetes Association, 2023):

1. **Red Blood Cells and Sugar:** Your red blood cells (RBCs) have hemoglobin, a protein that carries oxygen. When glucose (sugar) is in your bloodstream, it can attach (or "glycate") to the hemoglobin in red blood cells (i.e., the so-called, "glycated HbA1c").
2. **Lasts About Three Months:** Red blood cells typically live about 120 days. The A1c measurement reflects the percentage of hemoglobin that has glucose bound to it, providing an indication of the average blood sugar level over this lifespan.
3. **Why It's Important:**
    - For people with diabetes, the A1c test is a key indicator of how well blood sugar is being controlled over time.
    - Unlike daily blood sugar "glucose" checks, which can fluctuate based on meals, stress, or exercise, the A1c offers a long-term view.
4. **Typical Clinical Decision Cutoffs:**
    - An A1c below 5.7% is generally considered normal.
    - An A1c of 5.7–6.4% can indicate prediabetes.
    - An A1c of 6.5% or higher usually indicates diabetes.

By knowing their A1c, individuals and healthcare providers can adjust medications, diet, and lifestyle to maintain healthier glucose levels over the long term.

Step 1. Let us create a simulated SAS data set that can be replicated on any user system.

```
/*--------------------------------------------------------------*/
/* Step 1: Create a SIMULATED dataset (replace with your actual) */
/*--------------------------------------------------------------*/
data myKondData ;
  call streaminit(123456) ; * For reproducible randomness ;
  do ID = 1 to 100 ;
    * Reference (X) in the 4%-12% range ;
    X_Reference = 4 + 8 * rand( 'Uniform' ) ;

    * Introduce a small random bias (+/- ~4%) around the reference ;
    BiasFactor  = 1 + (rand( 'Uniform' ) * 0.08 - 0.04) ;
    Y_Candidate = X_Reference * BiasFactor ;
    diffyx = Y_Candidate - X_Reference ;
    xyave  = mean(X_Reference,Y_Candidate) ;
    output ;
  end ;
run ;
```

Code 1. SAS code for creating a simulated HbA1c Data Set.

Step 2. We create the data set with the medical decision points or cutoffs. Referring back to section 3.2.4, the clinical decision cutoffs or medical decision points (MDP) are entered in the following Code 2.

```
* medical decision points about X ;
data DSMDP_HbA1c;
  length MDPValue Context $50 ;
  input MDPValue Context $char50.;
datalines;
5.6 <=5.6=normal, >5.6-<6.5=pre-diabetes
6.5 >=6.5 = Type II diabetes mellitus
;
run;
```

Code 2. SAS code for defining medical decision points (MDP) or cutoffs about X.

Step 3. Imagine that our simulated study had the following variations in standard deviation units at each expected value/mean observed value, obtained from the old/reference study's instructions for use (IFU) or from its assay development phase, as follows:

Expected Value HbA1c %	Mean Observed Value HbA1c %	Standard Deviation HbA1c %
4.5	4.52	0.07
5.5	5.48	0.09
6.5	6.53	0.13
7.5	7.56	0.18

8.5	8.54	0.24

Table 1. HbA1c % expected value, mean observed value and standard deviation.

Enter the precision Table 1 values from the published X method into a SAS data set:

```
* Step 3. Enter table of precision values ;
data oldsys_repro ;
  infile datalines dlm="," dsd ;
  input ExpectedValue : 8.2
        MeanObserved  : 8.2
        SD            : 8.3 ;
Datalines ;
4.50,4.52,0.07
5.50,5.48,0.09
6.50,6.53,0.13
7.50,7.56,0.18
8.50,8.54,0.24
;
run;
```

Code 3. SAS code for precision values at different areas of the method X range.

We will utilize the A1c example above to exemplify the ATDzone SAS macro inputs. Table 1 shows the complete set of macro inputs. Importantly, the inputs include three required data sets that must be considered that include specific variable inputs within each set. The **three data sets** and **required variables** are:

1. ***dspi*** : This is a data set that includes titers or levels and averages, and standard deviations at each of those levels. Within this data set, there are two required variables:

   titervar : This is the variable name input of expected value (or assigned value) for each level found in ***dspi*** dataset. In our A1c example, this variable is called *ExpectedValue.*

   sdvar : This is the variable name input of the standard deviation for each level found in the *dspi* dataset. In the A1c example, this variable is called ***sd***.

2. ***dsmdp*** : This is the data set (ds) that include the medical decision points (mdp) or clinical cutoffs. Within this data set, there is one required variable:

   mdpvar : This is the medical decision point (or clinical cutoff) variable name found in the *dsmdp* data set. In our A1c example, this variable is called *MDPValue* as found in **Code 2**.

3. ***dsin*** : This is the main data set input name. It contains the X and Y methods and two derived variables in your data step (as shown in **Code 1** and relevant code block reshown below):

```
…
diffyx = Y_Candidate - X_Reference ;      * derived ;
xyave  = mean(X_Reference,Y_Candidate) ; * derived ;
…
```

Code 4. Required two derived variables for the difference (i.e., Y-X on y-axis) and average (i.e., (X+Y/2) on x-axis).

Table 1. ATDzone SAS macro inputs, description, and defaults.

ATDzone SAS Macro Inputs	Type: Description	Required or Optional	Default
dspi	**Data**: Package insert dataset including titers, averages, and standard deviations.	Required	---
titervar	**Variable**: Variable name of expected value (or assigned value) for each level found in *dspi* dataset	Required	---
sdvar	**Variable**: Variable name of the standard deviation for each level found in the dspi dataset	Required	---
dsmdp	**Data**: Medical decision point data about x	Required	---
mdpvar	**Variable**: Medical decision point variable name found in *dsmdp* data set	Required	---
dsin	**Data**: Main input data set.	Required	---
xyave	**Variable**: x-axis variable name (typically, mean of X and Y). Derive this in your data step (see **Code 4**).	Required	---
xyavelab	**Label**: x-axis variable label	Optional	<< empty set >>
difxy	**Variable**: y-axis variable name (typically, Y-X difference). Derive this in your data step (see **Code 4**).	Required	---
difxylab	**Label**: y-axis variable label	Optional	<< empty set >>
SDadjust	**Switch**: Establishing SD and percent CV for ATD Based on the Performance of the Old System online ref (PDF): https://www.fda.gov/media/73669/download?attachment on or about page 50, accessed on 21Feb2025	Optional	<< empty set >>
ols	**Switch**: 1 = fit ATD zone about OLS regression line, 0 = fit ATD zone about Y-axis=0	Optional	0
title1	**Title**: Main title at the top of the output graph.	Optional	<< empty set >>
title2	**Sub-Title**: Subtitle below the Main title.	Optional	<< empty set >>

ATDzone SAS Macro Inputs	Type: Description	Required or Optional	Default
`out`	**Data**: Output dataset name appending ATD zone boundaries	Optional	*ATDzoneOutBnd*
`out_atd`	**Data**: Output dataset name that contains numbers/percentages inside ATD Zone	Optional	*ATDzoneOutPct*
`graph`	**Switch**: 1 = output graph, 0 = omit graph	Optional	1
`GraphMin`	**Graph Parameter**: if graph is output, enter minimum x-axis value for output graph	Optional	System Algorithm
`GraphMax`	**Graph Parameter**: if graph is output, enter maximum x-axis value for output graph	Optional	System Algorithm
`GraphIncrement`	**Graph Parameter**: if graph is output, enter increment on x-axis	Optional	1
`BiasGraphMin`	**Graph Parameter**: if graph is output, enter minimum y-axis value for output graph	Optional	System Algorithm
`BiasGraphMax`	**Graph Parameter**: if graph is output, enter maximum y-axis value for output graph	Optional	System Algorithm
`BiasIncrement`	**Graph Parameter**: if graph is output, enter increment on y-axis	Optional	0.5
`ParmDecimalPlaces`	**Graph Parameter**: if graph is output, enter decimal places for values on graph	Optional	0.001
`dir_file`	**File**: RTF results file using SAS ODS	Required	---

ATD=Allowable Total Difference, SD=Standard Deviation, CV=Coefficient of Variation, OLS=Ordinary Least Squares, ODS=Output Delivery System.

4. Application Example

Using the HbA1c example in section 3.2, we demonstrate the ATDzone SAS macro as follows.

In your SAS session, run **Code 1-3** above. Then run the following SAS code:

```
%LET path = << insert your directory path for your system – no quotes >> ;

/*    ATD zone macro        */
%include "&path./atdzone.sas";
%atdzone(dspi    = oldsys_repro
,titervar        = ExpectedValue
,sdvar           = std
,dsmdp           = DSMDP_HbA1c
,mdpvar          = MDPValue
,dsin            = myKondData
,xyave           = xyave
,xyavelab        = "Avg of X and Y"
,difxy           = diffyx
,difxylab        = "Difference between Y and X"
,SDadjust        = 0 /* 0=no SD adjustment, 1=SD adjustment */
,ols             = 0 /* 0=about zero line (default), 1=about regression line */
,title1          = "%HbA1c Example (about the zero line)"
,title2          = "PharmaSUG 2025, San Diego"
,out             = atd
,out_atd         = atd_z
,graph           = 1    /* 0=no graph, 1=graph */
,GraphMin        =      /* optional */
,GraphMax        =      /* optional */
,GraphIncrement  = 0.5  /* optional */
,BiasGraphMin    = -2   /* optional */
,BiasGraphMax    = +2   /* optional */
,BiasIncrement   = 0.1  /* optional */
,ParmDecimalPlaces = 0.001 /* optional */
,dir_file= &path./ATD_Test_AboutZero_NoSDadj.rtf
) ;
```

Code 5. ATDzone SAS macro call using the HbA1c example in section 3.2.

5. Results

The results from running the **Code 1-3** and **Code 5** yield the following ATD zone graph:

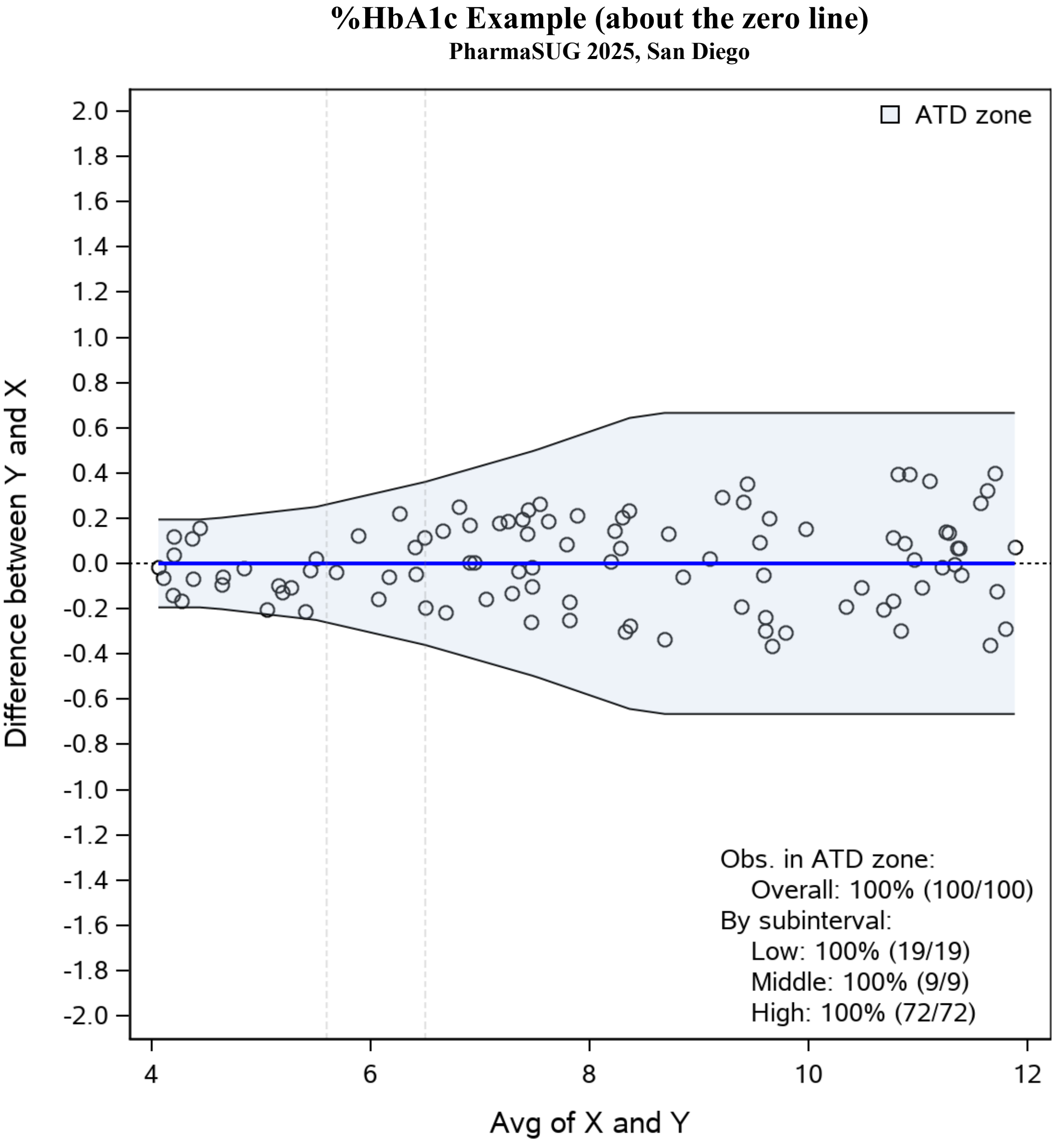

Figure 1. Allowable Total Difference **(**ATD) zone graph of HbA1c simulated data from section 3.2 after running **Code 1-3** and **Code 5**.

Interpretation

Figure 1 shows 100% of values fall within the low, middle and high sub-intervals and 100% of observations fall within the ATD zone, indicating that the new/test method Y performance is within the variation of the old/reference method X, thus providing evidence for equivalence between methods. Regulatory agencies may require greater than 95% performance within each region to declare equivalence between the two methods.

6. Conclusion

The SAS Macro streamlines ATD zone construction, enhancing compliance with FDA guidelines for assay migration studies. This tool aids researchers in robustly evaluating measurement comparability, supporting regulatory submissions.

REFERENCES

1. Altman DG and Bland JM (1983). “Measurement in medicine: the analysis of method comparison studies.” *The Statistician*; 32, 307-317.
2. American Diabetes Association (2023). “2. Classification and diagnosis of diabetes: standards of care in diabetes—2023”. Diabetes care, 46, S19-S40.
3. Bland JM and Altman DG (1986). “Statistical methods for assessing agreement between 2 methods of clinical measurement.” *The Lancet*; 307–310.
4. Clarke WL, Cox D, Gonder-Frederick LA, Carter W, Pohl SL (1987). "Evaluating Clinical Accuracy of Systems for Self-Monitoring of Blood Glucose"*. Diabetes Care. 10 (5):* 622–*628.* doi*:*10.2337/diacare.10.5.622*.* PMID 3677983*.* S2CID 26388964*.*
5. CLSI. Estimation of Total Analytical Error for Clinical Laboratory Methods; Approved Guideline (2003). CLSI document EP21-A. Wayne, PA: Clinical and Laboratory Standards Institute.
6. CLSI. Measurement Procedure Comparison and Bias Estimation Using Patient Samples; Approved Guideline—Third Edition (2013). CLSI document EP09-A3. Wayne, PA: Clinical and Laboratory Standards Institute.
7. FDA. Assay Migration Studies for In Vitro Diagnostic Devices – Guidance for Industry and FDA Staff. April 25, 2013.
8. Krouwer JS, Monti KL (1995). “A Simple, Graphical method to Evaluate Laboratory Assays.” *Eur J Clin Chem Clin Biochem*. 33:525-527
9. Krouwer JS (2008). “Why Bland-Altman plots should use X, not (Y+X)/2 when X is a reference method.” *Stat Med*. 27:778–80.
10. Linnet K (1993). “Evaluation of Regression Procedures for Methods Comparison Studies.” *Clin. Chem.*, 39, 3, 424-432.
11. Parkes JL, Slatin SL, Pardo S, & Ginsberg BH (2000). “A new consensus error grid to evaluate the clinical significance of inaccuracies in the measurement of blood glucose.” *Diabetes care*, *23*(8), 1143-1148.
12. Passing H & Bablok W (1983). “A new biometrical procedure for testing the equality of measurements from two different analytical methods. Application of linear regression procedures for method comparison studies in clinical chemistry, Part I.” De Gruyter, 709-720.

ACKNOWLEDGMENTS

The authors thank Enrique Marino and Yang Bai for their code implementation suggestions and validation.

CONTACT INFORMATION

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

Name: Jesse A. Canchola
Company: Roche Molecular Systems, Inc
Address: 4300 Hacienda Drive
City, State ZIP: Pleasanton, California, 94588
E-mail: Jesse.Canchola@Roche.Com
Online Macro Directory Download: http://bit.ly/4hHpKfq

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are trademarks of their respective companies.

APPENDIX A: FULL SAS MACRO CODE

```
/* ***************************************************************************
Program:          atdzone.sas

Purpose:          Macro to create ATD zone per CLSI EP21
                  https://clsi.org/standards/products/method-evaluation/documents/ep21/
                  and FDA assay migration study document

Input:            Needs dataset of pairwise results
                  1. average: (X+Y)/2 and
                  2. difference (Y-X) between 2 assays: X=Reference, Y=Test

Output:           New dataset with percentage of points inside per zone
                                    Allowable Total Difference plot with cutoffs at each
                  relevant Medical Decision Point

Created by:            Jesse A. Canchola (JAC) and Enrique Marino (E-M)
Creation Date:    14Aug2014 by JAC and E-M
Modify Date:       16June2016 by JAC
Modify Date:      23Jan2025 by JAC

DEFINE MACRO PARAMETERS
    dspi         : data: package insert dataset about x
                 should include (1) titers, (2) average result for each titer,
                 and (3) sd for each titer
    titervar     : variable: variable name of expected value (or assigned value) for each level
                 found in dspi dataset
    sdvar        : variable: variable name of the standard deviation for each level
                 found in dspi dataset
    dsmdp        : data: medical decision point data about x
    mdpvar       : variable: medical decision point variable name found in dsmdp data set
    dsin         : data: input dataset
    xyave        : x-axis variable (typically, mean of X and Y)
    xyavelab     : x-axis variable label
    difxy        : y-axis variable (typically, Y-X difference)
    difxylab     : y-axis variable label
    SDadjust     : Establishing SD and percent CV for ATD Based on the
                       Performance of the Old System
                       online ref (PDF): https://www.fda.gov/media/73669/download?attachment
                       on or about page 50, accessed on 21Feb2025
    ols          : 1 = fit ATD zone about OLS regression line, 0 = fit ATD zone about Y = 0
    title1       : main title of graph
    title2       : subtitle of graph
    out          : output dataset name appending atd zone boundaries
    out_atd      : output dataset name that contains numbers/percentages inside ATD Zone
    graph        : 1 = output graph, 0 = omit graph
    GraphMin       : (optional) if graph is output, enter minimum x-axis value for output graph
    GraphMax       : (optional) if graph is output, enter maximum x-axis value for output graph
    GraphIncrement : (optional) if graph is output, enter increment on x-axis
    BiasGraphMin   : (optional) if graph is output, enter minimum y-axis value for output graph
    BiasGraphMax   : (optional) if graph is output, enter maximum y-axis value for output graph
    BiasIncrement  : (optional) if graph is output, enter increment on y-axis
    ParmDecimalPlaces: (optional) if graph is output, enter decimal places for values on graph
    dir_file       : RTF results file using SAS ODS
 */

%macro atdzone(
 dspi                =
,titervar            =
,sdvar               =
,dsmdp               =
,mdpvar              =
,dsin                =
,xyave               =
,xyavelab            =
,difxy               =
,difxylab            =
,SDadjust            =
,ols                 =
,title1              =
,title2              =
,out                 =
,out_atd             =
,graph               =
,GraphMin            =
,GraphMax            =
,GraphIncrement      =
,BiasGraphMin        =
,BiasGraphMax        =
,BiasIncrement       =
,ParmDecimalPlaces =
,dir_file          =
) ;

title1 &title1. ;
title2 &title2. ;

* defaults section ;
%if &SDadjust.          = %then %let SDadjust          = 0     ; * no SD adjustment ;
%if &ols.               = %then %let ols               = 0     ; * no ols fit ;
%if &out.               = %then %let out               = ATDzoneOutBnd ;
%if &out_atd.           = %then %let out_atd           = ATDzoneOutPct ;
%if &graph.             = %then %let graph             = 1     ;
%if &GraphIncrement.    = %then %let GraphIncrement    = 1     ;
%if &BiasIncrement.     = %then %let BiasIncrement     = 0.5   ;
%if &ParmDecimalPlaces. = %then %let ParmDecimalPlaces = 0.001 ;
* end defaults section ;

%if &ols. = 1 %then
  %do ;
         *        Regression to build line   ;
        proc reg data=&dsin. ;
                 model &difxy.  = &xyave. / clb ;
                 ods output ParameterEstimates=PE_avg ;
        run ;
        data _null_ ;
                 set PE_avg ;
                 if _n_ = 1 then
        call symput("int", strip(put(round(estimate, &ParmDecimalPlaces.), best6.))) ;
                 else
        call symput("slp", strip(put(round(estimate, &ParmDecimalPlaces.), best6.))) ;
        run ;
        %end ;
  %else %do ;
         *        Build line around Y = 0 line       ;
        data _null_ ;
                 call symput("int", 0.0) ;
                 call symput("slp", 0.0) ;
        run ;
  %end ;

  proc reg data=&dsin. outest=EDF(rename=(_EDF_=sddf)) edf ;
                 model &difxy.  = &xyave. / clb ;
  run ;
  data _null_ ;
             set EDF ;
                 call symputx("sddf",sddf) ;
  run ;

*        determine min and max on x-axis from dsin            ;
proc sql noprint ;
        select min(&xyave. ) into: x_min
        from &dsin. ;
quit ;
proc sql noprint ;
        select max(&xyave. ) into: x_max
        from &dsin. ;
quit ;

*        extrapolate outside repro data              ;
data first ;
        set &dspi. ;
        if _N_ = 1 and (&titervar. > &x_min.) ;
        if (&titervar. > &x_min.) then do ;
                 &titervar. = floor(&x_min.) ;
                 mean = floor(&x_min.) ;
        end ;
run ;
data last ;
        set &dspi. end=eof ;
        if eof and (&titervar. < &x_max.) ;
        if (&titervar. < &x_max.) then do ;
                 &titervar. = ceil(&x_max.) ;
                 mean = ceil(&x_max.) ;
        end ;
run ;
data first_dspi_last ;
        set first &dspi. last ;
run ;

*        calculate lower and upper bounds at each repro point ;
```

```
data atd ;
         set first_dspi_last ;

         yhat = &int. + &slp. * &titervar. ;

         * Establishing SD and percent CV for ATD Based on the
         * Performance of the Old System ;
         * online ref (PDF): https://www.fda.gov/media/73669/download?attachment
         * on or about page 50, accessed on 21Feb2025 ;
         chisq = cinv(0.05,&sddf.) ;
         f = ( 1 - 1 / (4 * &sddf.) )**(-1) * SQRT(&sddf. / chisq) ;

         %if &SDadjust. = 1 %then
           %do ;
             atd_upper = yhat + 1.96 * sqrt(2) * &sdvar. * f ;
             atd_lower = yhat - 1.96 * sqrt(2) * &sdvar. * f ;
           %end ;
         %else
      %do ;
             atd_upper = yhat + 1.96 * sqrt(2) * &sdvar. ;
             atd_lower = yhat - 1.96 * sqrt(2) * &sdvar. ;
           %end ;
run ;

*        calculate number of points in appended dspi dataset ;
proc sql noprint ;
         select count(*) into: atdobs
         from first_dspi_last ;
quit ;

*        calculate the slopes of upper and lower bounds at each titer level ;
proc iml ;
         use atd ;
         read all var{atd_upper} into y_h ;
         read all var{atd_lower} into y_l ;
         read all var{&titervar.} into x ;
         close atd ;

         S_h = j(&atdobs., 1, .) ;
         S_l = j(&atdobs., 1, .) ;

         do i=1 to (&atdobs. - 1) ;
                  S_h[1] = 0 ;
                  if (x[i+1] - x[i])=0 then S_h[i]=. ;
                  else S_h[i+1] = (y_h[i+1] - y_h[i])/(x[i+1] - x[i]) ;
         end ;

         do i=1 to (&atdobs. - 1) ;
                  S_l[1] = 0 ;
                  if (x[i+1]-x[i])= 0 then S_l[i] = . ;
                  else S_l[i+1] = (y_l[i+1] - y_l[i])/(x[i+1] - x[i]) ;
         end ;

         create S_hi from S_h ;
         append from S_h ;

         create S_lo from S_l ;
         append from S_l ;
quit ;

*        append slopes to each point         ;
data atd_S_hi_lo ;
         merge atd S_hi(rename=COL1=slope_upper) S_lo(rename=COL1=slope_lower) ;
run ;

*        calculate number of observations in dataset ;
proc sql noprint ;
         select count(*) into: obs_data
         from &dsin. ;
quit ;

*        calculate the upper and lower bounds of the atd zone line ;
proc iml ;
         use &dsin. ;
         read all var{&xyave.} into z ;
         close &dsin. ;

         use atd_S_hi_lo ;
         read all var{&titervar.} into a ;
         read all var{atd_upper} into w ;
         read all var{atd_lower} into v ;
```

```
        read all var{slope_upper} into slope_h ;
        read all var{slope_lower} into slope_l ;
        close atd_S_hi_lo ;

        atd_u = j(&obs_data., 1, .) ;
        atd_l = j(&obs_data., 1, .) ;

        do i=1 to &obs_data. ;
                do j=1 to (&atdobs. - 1) ;
                        if (z[i] <= a[j+1]) & (z[i] > a[j]) then ;
                                do ;
                                        atd_u[i] = w[j] + slope_h[j+1] * (z[i] - a[j]) ;
                                        atd_l[i] = v[j] + slope_l[j+1] * (z[i] - a[j]) ;
                                end ;
                end ;
        end ;
        create Yn_upper from atd_u ;
        append from atd_u ;

        create Yn_lower from atd_l ;
        append from atd_l ;
quit ;

data dsin_yu ;
        merge &dsin. Yn_upper (rename=(COL1=yhat_upper)) ;
run ;
data dsin_yu_yl ;
        merge dsin_yu Yn_lower (rename=(COL1=yhat_lower)) ;
        if &difxy.  le yhat_upper then below=1 ; else below=0 ;
        if &difxy.  ge yhat_lower then above=1 ; else above=0 ;
        inside=below*above ;
run ;

*       calculate percent inside ATD zone ;
*       overall ;
proc sql ;
        create table overall as
        select "Overall" as category, sum(inside) as n, count(inside) as tot, mean(inside)*100 as pct
        from dsin_yu_yl ;
quit ;
data _null_ ;
        set overall ;
        call symputx("n_all", n) ;
        call symputx("tot_all", tot) ;
        call symputx("pct_all", put(pct, best4.)) ;
run ;

*       calculate percent inside ATD zone by subinterval ;
proc iml ;
        use &dsmdp. ;
        read all var{&mdpvar.} into mdp ;
        close &dsmdp. ;

        call symputx("low_mid", mdp[1]) ;
        call symputx("mid_high", mdp[2]) ;
quit ;

*       low                ;
proc sql ;
        create table low as
        select "Low" as category, sum(inside) as n, count(inside) as tot, mean(inside)*100 as pct
        from dsin_yu_yl
        where &xyave.  le &low_mid. ;
quit ;
data _null_ ;
        set low ;
        call symputx("n_low", n) ;
        call symputx("tot_low", tot) ;
        call symputx("pct_low", put(pct, best4.)) ;
run ;

*       middle    ;
proc sql ;
        create table middle as
        select "Middle" as category, sum(inside) as n, count(inside) as tot, mean(inside)*100 as pct
        from dsin_yu_yl
        where (&low_mid. < &xyave.  < &mid_high.) ;
quit ;
data _null_ ;
        set middle ;
        call symputx("n_mid", n) ;
        call symputx("tot_mid", tot) ;
        call symputx("pct_mid", put(pct, best4.)) ;
run ;

*       high      ;
proc sql ;
        create table high as
        select "High" as category, sum(inside) as n, count(inside) as tot, mean(inside)*100 as pct
        from dsin_yu_yl
        where &xyave.  gt &mid_high. ;
quit ;
data _null_ ;
        set high ;
        call symputx("n_high", n) ;
        call symputx("tot_high", tot) ;
        call symputx("pct_high", put(pct, best4.)) ;
run ;

data &out_atd. ;
        length category $9 ;
        set low middle high overall ;
run ;

*       create dataset to plot the ATD zone ;
proc iml ;
        *       read data ;
        use dsin_yu_yl ;
        read all var {&xyave. } into x ;
        read all var {&difxy. } into y ;
        close dsin_yu_yl ;

        *       find min and max of x values ;
        xmin = min(x) ;
        xmax = max(x) ;

        *       linear helper function ;
        start L(x) ;
                *       intercept ;
                a = &int. ;
                *       slope ;
                b = &slp. ;
                return(a + b*x) ;
        finish ;

        *       write vars LX and LY to endLine data set ;
        Lymin = L(xmin) ;
        Lymax = L(xmax) ;

        create endLine var {xmin xmax Lymin Lymax} ;
        append ;
        close endLine ;
quit ;

*       sort by x-variable ;
proc sort data=dsin_yu_yl ;
        by &xyave.  ;
run ;

*       output dataset ;
data &out. ;
        set dsin_yu_yl endLine &dsmdp. ;
run ;


%if &graph. = 1 %then
  %do ;
    ods _ALL_ close ;
    ods escapechar = '~' ;
    %modstyle(name = markers,
              parent = listing,
              type = CLM,
                       colors = black,
              markers = circle) ;

    ods graphics / reset=all width=550px height=550px border=off imagename="PNG" imagefmt=png ;
    ods rtf file="&dir_file." bodytitle notoc_data nogtitle style=markers ;
    ods noptitle ;

    title1 bold justify=center height=12pt f="Times"  &title1. ;
    title2 bold justify=center height=10pt f="Times"  &title2. ;
    proc sgplot data=&out. ;
```

```
            *    scatter ;
            scatter x=&xyave.  y=&difxy. ;

            *    ATD zone ;
            series x=&xyave.  y=yhat_upper ;
            series x=&xyave.  y=yhat_lower ;
            band x=&xyave.  lower=yhat_lower upper=yhat_upper / transparency=.75 name="ATD" legendlabel="ATD
zone" ;

            *    Y = 0 line ;
            lineparm x=0 y=0 slope=0 / lineattrs=(color=black pattern=2) ;

            *    ATD Zone about this Line   ;
            vector x=xMax y=Lymax /xorigin=xMin yorigin=Lymin noarrowheads lineattrs= (thickness=2.5 color=blue
pattern=1) ;

            *    Reference lines determining Low, Middle, High intervals  ;
            refline &low_mid.  / axis=x lineattrs=(pattern=3) transparency=.75 ; *label="&low_mid."
labelloc=inside labelpos=min ;
            refline &mid_high. / axis=x lineattrs=(pattern=3) transparency=.75 ; *label="&mid_high."
labelloc=inside labelpos=min ;

%if &BiasGraphMin. eq and &BiasGraphMax. eq
    %then %do ;
                  YAXIS LABEL = &difxylab. ;
          %end ;
         %else %do ;
                  YAXIS LABEL = &difxylab.
                  VALUES = (
                            %if &BiasGraphMin. = %then
                                %do ;
                                  &BiasGraphMin1.
                                                                      %end ;
                                                                      %else %do ;
                                  &BiasGraphMin.
                                                                      %end ;
                                                                      TO
                            %if &BiasGraphMax. = %then
                                                                      %do ;
                                  &BiasGraphMax1.
                                                                      %end ;
                                                                      %else %do ;
                                  &BiasGraphMax.
                                                                      %end ;
                                BY &BiasIncrement.
                            ) ;
         %end ;

%if &GraphMin. eq  and &GraphMax. eq
    %then %do ;
                  XAXIS LABEL = &xyavelab.  ;
          %end ;
         %else %do ;
                  XAXIS LABEL = &xyavelab.
                  VALUES = (
                            %if &GraphMin. = %then
                                %do ;
                                  &GraphMin1.
                                                                      %end ;
                                                                      %else %do ;
                                  &GraphMin.
                                                                      %end ;
                                                                      TO
                            %if &GraphMax. = %then
                                                                      %do ;
                                  &GraphMax1.
                                                                      %end ;
                                                                      %else %do ;
                                  &GraphMax.
                                                                      %end ;
                                BY &GraphIncrement.
                            ) ;
          %end ;

          inset "Obs. in ATD zone:"
                    "    Overall: &pct_all% (&n_all./&tot_all.)"
                    "By subinterval:"
                    "    Low: &pct_low% (&n_low./&tot_low.)"
                    "    Middle: &pct_mid% (&n_mid./&tot_mid.)"
                    "    High: &pct_high% (&n_high./&tot_high.)" / noborder textattrs=(color=black size=9px)
position=bottomright noborder ;
```

```
           keylegend "ATD" / noborder location=inside position=topright ;
    run ;
    ods rtf close ;
  %end ;
%mend atdzone ;
```

*************************** end macro code ************************* ;

< End of Document >