Paper #60-2022

Customizable SAS Graphs for Bias Analysis

Natasha Oza and Jesse A. Canchola

Roche Diagnostics Solutions, Pleasanton, California, USA

ABSTRACT

Graphical methods for bias analysis are well characterized (Bland and Altman, 1983). The papers by Fernandez and Fernandez (2009) and Johnson and Waller (2018) go into some detail on both the theory and SAS code for creating bias plots. However, the graphical coding is basic and not necessarily tailorable to the user requirements. We introduce the BIASPLOTS SAS macro that produces highly customizable camera-ready graphs that contain details of the bias plot.

INTRODUCTION AND BACKGROUND

Bias plots are a visual method for evaluating differences between two measurement methods on the same "item" (e.g., a blood draw (1) testing a particular target such as HIV, HBV, HCV, SARS-CoV-2 and so on or (2) measuring body fat, blood pressure, etc.) Typically, the assessment is how similar or different a new technique or a new instrument is at measuring that item, both in magnitude of agreement and direction of error (i.e., bias), as compared to the currently accepted technique or instrument (i.e., a reference or "gold" standard) or a comparator.

There are at least two types of standard bias plots. The first type of standard bias plot is due to Bland and Altman (1,2) and is called the "Bland-Altman bias plot". This type of plot graphs (Y-X) on the y-axis vs. their average (Y+X)/2 (Figure 2). The second type of bias plot is the "Krouwer bias plot" (4) measuring the difference between the test (i.e., Y=new) and reference (i.e., X=older or standard) method, that is, Y-X, graphed against the reference method, X. For this type of bias plot, the difference "(Y-X)" is on the y-axis and the reference "X" is on the x-axis (Figure 1). Table 1 shows eight of the possible bias plots. The current BIASPLOTS SAS macro produces plot types 1 (Bland-Altman) and 2 (Krouwer) with future work towards incorporating the rest in Table 1.

Table 1. Eight Different Bias Plots

Plot Type	X-axis	Y-axis	Notes
1	$\frac{(X+Y)}{2}$	Y - X	Bland-Altman bias plot (default).
2	X	Y - X	Krouwer bias plot.
3	X	$\frac{(Y-X)}{X}$	Proportion change from X vs. X.
4	$\frac{(X+Y)}{2}$	$\frac{(Y-X)}{X}$	Proportion change from X vs. mean of X,Y.
5	rank(X)	Y-X	X,Y difference vs. rank of X. Useful for skewed distributions.
6	rank(X)	$\frac{(Y-X)}{X}$	Proportion change from X vs. rank of X. Useful for skewed distributions.
7	$\sqrt{X \cdot Y}$	$\frac{Y}{X}$	Ratio of Y and X vs. square root of the X,Y product. Useful when transforming to normality and reducing heteroskedasticity.
8	$\frac{(X+Y)}{2}$	$\frac{(Y-X)}{\left[\frac{(X+Y)}{2}\right]}$	Mean-adjusted X,Y difference vs. mean of X,Y.

PROGRAMMING

This BIASPLOTS SAS macro produces the following:

1. Estimates of OLS regression parameters and associated CIs;

2. Plot types 1 and 2 (Table 1) [note: the rest will be integrated in the next version];

3. Confidence bands are included at the stated confidence level, where alpha is the significance level with default set at 0.05.

The entire source code set is shown in Appendix A. The example macro call is included in Appendix B. For convenience, the user may download the macro and supporting documents and any updates at this online link: https://bit.ly/3SD01tk

METHODS

There are four main steps to successfully run the BIASPLOTS SAS macro as follows:

Step 1: The following data variables are required. Table 2, below, details all the essential variables involved.

a. Input data name
b. Output data name
c. Unique identifier
d. 'Independent' or 'X' variable
e. 'Dependent' or 'Y' variable

Note: The macro will automatically keep only valid paired data and will exclude any missing data using pairwise deletion.

Step 2: Load the BIASPLOTS.sas macro into your session.

```
%include "c:/sasmacros/biasplots.sas" ;
```

Step 3: Enter BIASPLOTS SAS macro inputs (see later example and **Appendix B**).

```
ods noproctitle ;
ods rtf file = "c:/sasmacros/output/YourOutput.rtf" gtitle ;
%biasplots(dsin=example
,     dsout= example_Out
,     title1= "title1"
,     title2= "title2"
,     title3=
,     sampleid= myid
,     iter= 1
,     xvar= x1
,     xlab= "x-axis"
,     yvar= y1
,     ylab= "y-axis"
,     diffxy= logdif
,     diffxylab= "difference label"
,     meanxy= logave
,     meanxylab= "avg label"
,     alpha= 0.05
,     GraphMin=
```

```
,     GraphMax=
,     GraphIncrement=
,     BiasGraphMin=
,     BiasGraphMax=
,     BiasIncrement=
,     ParmDecimalPlaces= 0.001) ;
ods rtf close ;
```

Step 4: Run your SAS code.

As indicated in the steps above, Table 2, shows the BIASPLOTS SAS macro inputs with parameter inputs, their descriptions and inputs and defaults. There are five inputs that are required. The input data set, *Dsin,* the outpout data set, *Dout,* the unique observation identifier, *SampleID*, the X variable*, Xvar* and the Y-vairable*, Yvar.*

The rest of the inputs are optional with either defaults or are calculated if the BIASPLOTS SAS macro does not "see" the variable for *diffxy*, the X-Y difference or *meanxy*, the mean of X and Y.

Parameter Number	Macro Parameter	Parameter Description	Specifications / Defaults
1	Dsin	SAS input data set	**required**
2	Dsout	SAS output data set	**required**
3	title1	main title	optional
4	title2	first subtitle	optional
5	title3	second subtitle	optional
6	Sampleid	unique row observation identifier	**required**
7	Iter	iteration/analysis number	optional
8	Xvar	“reference” or “independent variable	**required**
9	Xlab	label for xvar	optional
10	Yvar	“test” or “dependent” variable	**required**
11	Ylab	label for ylab	optional
12	Diffxy	difference between y and x	optional / calculated
13	Diffxylab	y-x difference label	optional
14	Meanxy	average of x and y	optional / calculated
15	Meanxylab	label for the mean of X,Y	optional
16	Alpha	significance level	optional / 0.05
17	GraphMin	minimum value for output graph	optional / SGPLOT auto-determined
18	GraphMax	maximum value for output graph	optional / SGPLOT auto-determined
19	GraphIncrement	increment on x- and y-axes	optional / 1.0
20	BiasGraphMin	minimum y-axis value for output graph	optional / SGPLOT auto-determined
21	BiasGraphMax	maximum y-axis value for output graph	optional / SGPLOT auto-determined
22	Biasincrement	increment on y-axis	optional / SGPLOT auto-determined
23	ParmDecimalPlaces	the number of decimal places for outputted values	optional / 0.001

Table 2. BIASPLOTS SAS macro variable inputs.

EXAMPLES

Next, example data are generated in the SAS code shown to the right. Deal, Pate, and El Rouby (2009) generate a data set as follows. For user convenience, the authors have added fixed randomization seeds in order to produce the same data set each time the code is run. The full code for this example can be found in **Appendix B**. The results below show the results of the execution of the BIASPLOTS SAS macro.

```
%LET SEED1 = 726453819 ;
%LET SEED2 = 289546394 ;
%LET SEED3 = 923847362 ;
ods listing close;
data Example ;
 do MyId = 1 to 200 ;
  Value = UNIFORM(&SEED1.) * 10 ;
  x1 = Value + NORMAL(&SEED2.) ;
  y1 = Value + NORMAL(&SEED3.) ;
  output ;
 end ;
 drop Value ;
run ;
```

Figures 1A and 1B show the Bland-Altman bias plots. Figure 1A captures the lower limit, bias, upper limit, sample size, mean difference and 95% confidence interval (CI) for the mean difference inside the graph. Figure 1B captures the lower limit, bias, upper limit, ordinary least squares (OLS) regression line with 95% CI for the slope of the line.

Figures 2A and 2B show the Krouwer bias plots. Similar to Figures 1A and 1B, respectively, Figure 2A captures the lower limit, bias, upper limit, sample size, mean difference and 95% confidence interval (CI) for the mean difference inside the graph. Figure 2B captures the lower limit, bias, upper limit, ordinary least squares (OLS) regression line with 95% CI for the slope of the line.

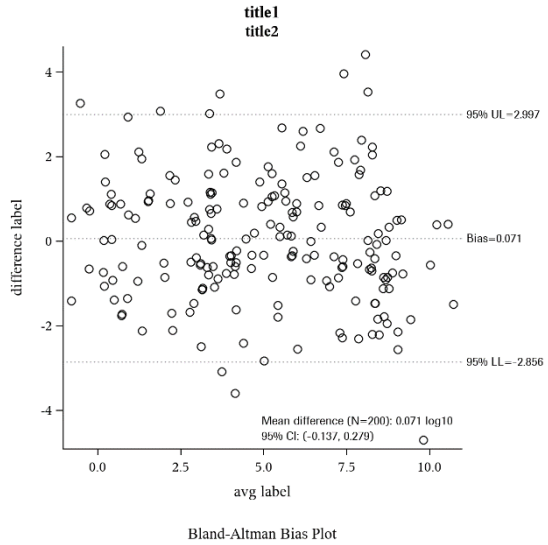
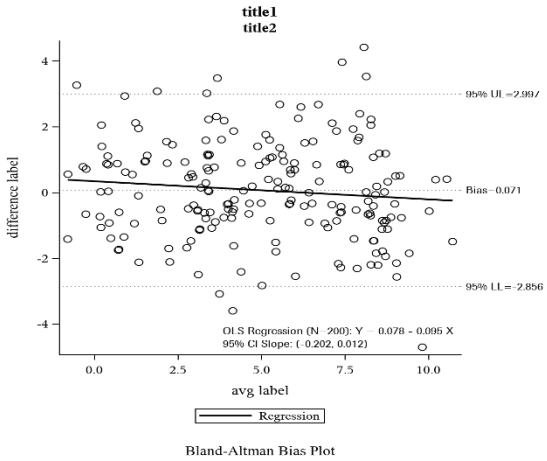
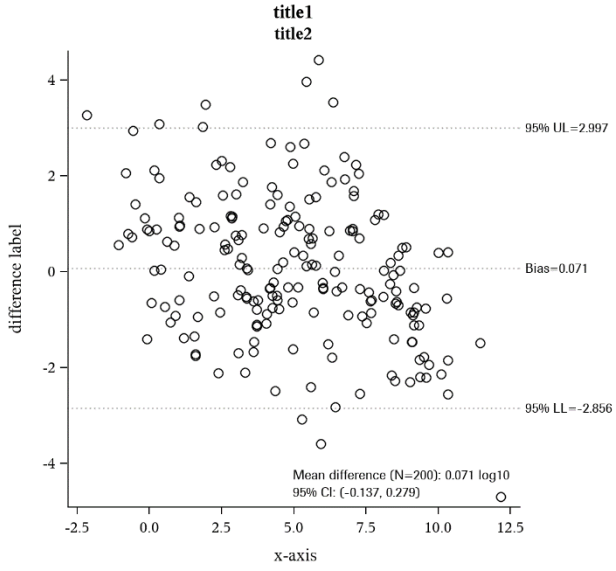


Figure 1A: This Bland-Altman bias plot captures the lower limit, bias, upper limit, sample size, mean difference and 95% CI for the mean difference inside the graph.



Bland-Altman Bias Plot

Figure 1B: This Bland-Altman bias plot captures the lower limit, bias, upper limit, sample size, ordinary least squares regression line with 95% CI for the slope



Krouwer Bias Plot

Figure 2A: This Krouwer bias plot captures the lower limit, bias, upper limit, sample size, mean difference and 95% CI for the mean difference inside the graph.

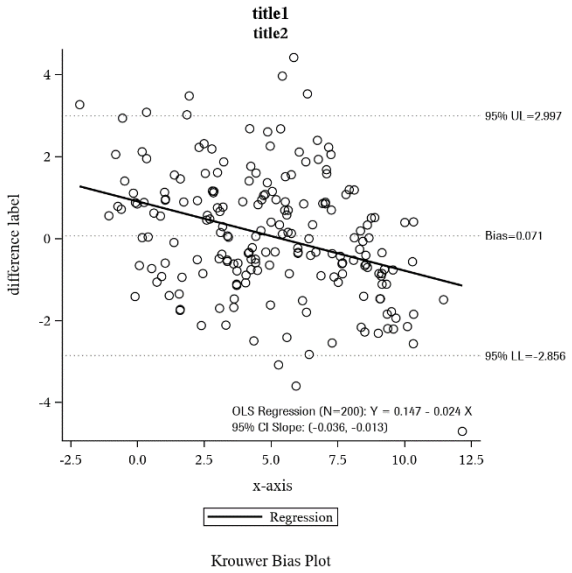


Figure 2B: This Krouwer bias plot captures the lower limit, bias, upper limit, sample size, and the ordinary least squares regression line with 95% CI for the slope.

DISCUSSION

The bias plots produced by the BIASPLOTS SAS macro produce the mean difference and associated 95% CI (Figures 1A and 2A). The validity of the mean difference comes into play if the nature of the bias is none, constant or proportional. If the bias across the range of the two methods is none or constant, then one mean difference value can be used to characterize the bias. However, if the bias is proportional (i.e., lower at the higher end and higher at the lower end or vice versa), then the use of a single value to characterize bias between the two methods is questionable. To test for zero or constant bias, the 95% confidence interval for the slope of the ordinary least squares (OLS) regression line should include zero. If the slope CI excludes zero, there is evidence for non-constant (i.e., proportional) bias. Figures 1B and 2B show the OLS regression lines for the Bland-Altman and Krouwer bias plots, respectively. The confidence interval for the slope in the Bland-Altman plot in Figure 1B includes zero (Slope estimate: -0.095, 95% CI for slope: -0.202, 0.012). Therefore, it can be concluded that the mean difference between the two methods as characterized by the Bland-Altman plot is sufficient to represent a constant bias value. Contrast this with the results for the Krouwer bias plot OLS regression results. There, the slope estimate was -0.024 with 95% CI from -0.036 to -0.013, indicating possible proportional or non-constant bias. One explanation is that the Bland-Altman represents a “smoothing” of the two methods when the difference (Y-axis) is plotted against the average (X-axis). One may even posit that the Krouwer method is useful when comparing a test or newer method to an actual reference method (e.g., an assay that has regulatory approval) whereas the Bland-Altman method is useful when comparing the test or newer method to an “imperfect” comparator method (e.g., a lab-developed assay; aka Lab-Developed Test or LDT). Given that the “lowest common denominator” in both Bland-Altman and Krouwer methods is to assume that both methods have error, the Bland-Altman bias method with its smoothing effects are set as the default. The reader should note the “animated” discussion, and perhaps controversy, regarding only using one or the other of the methods (5,6,7). At this time, the Bland-Altman remains the standard bias plot (and is the default), quickly followed by the Krouwer bias plot.

CONCLUSION

Unlike previous SAS bias plot macros, the BIASPLOTS SAS macro produces camera-ready Bland-Altman and Krouwer bias plots. The downloadable code is fully customizable at the PROC SGPLOT level and makes for an efficient and effective analysis experience.

REFERENCES

1. Altman DG, Bland JM. Measurement in medicine: the analysis of method comparison studies. The Statistician 1983; 32, 307-317
2. Bland JM, Altman DG. Statistical methods for assessing agreement between 2 methods of clinical measurement.*The Lancet* 1986; 307–310.
3. Deal AM, Pate VW, El Rouby S (2009). "A SAS® Macro for Deming Regression". Proceedings of the PharmaSUG 2009 Conference. Portland, OR. Paper CC-014.
4. Krouwer Jan S, Monti Katherine L. A Simple, Graphical method to Evaluate Laboratory Assays. Eur J Clin Chem Clin Biochem 1995; 33:525-527
5. Krouwer JS. Why Bland-Altman plots should use X, not (Y+X)/2 when X is a reference method. Stat Med. 2008;27:778–80.
6. M.A. Mansournia, R.Waters, M. Nazemipour et al. Bland-Altman methods for comparing methods of measurement and response to criticisms. Global Epidemiology 3 (2021) 100045.
7. Bland JM, Altman DG. Comparing methods of measurement: why plotting difference against standard method is misleading. Lancet. 1995;8982:1085–7.

CONTACT INFORMATION

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

Natasha Oza
Roche Molecular Systems, Inc.
4300 Hacienda Drive
Pleasanton, CA 94588
E-mail: Natasha.Oza@Roche.com

APPENDIX A: BIASPLOTS SAS MACRO CODE

```
* ************************************************************** ;
* SAS Macro BIASPLOTS
*
* This macro produces the following:
* 1. Estimates of OLS regression parameters and associated &ci. CIs
* 2. Various bias scatterplots
* 3. Confidence bands are included at the stated confidence level:
*   (1-alpha)*100% where alpha = significance level default set at 0.05
*
* Macro Inputs:
*
* dsin: enter input data set name
* dsout: enter output data set name
* title1: enter your main title
* title2: enter your subtitle
* title3: enter your subtitle
* sampleid: enter unique observation id
* iter: iteration/analysis number
* xvar: enter "reference" or "independent" variable
* xlab: enter label for xvar
* yvar: enter "test" or "dependent" variable
* ylab: enter label for yvar
* diffxy: enter y-x difference (if not entered it will be calculated for you
*                               using xvar and yvar)
* diffxylab: enter diffxy variable label
* meanxy: enter x,y average (if not entered it will be calculated for you
*                            using xvar and yvar)
* meanxylab: enter meanxy variable label
* alpha: enter significance level
* GraphMin: enter minimum x-axis value for output graph
* GraphMax: enter maximum x-axis value for output graph
* GraphIncrement: enter increment on x-axis
* BiasGraphMin: enter minimum y-axis value for output graph
* BiasGraphMax: enter maximum y-axis value for output graph
* BiasIncrement: enter increment on y-axis
* ParmDecimalPlaces: enter decimal places for outputted values
*
* Macro Defaults
* %macro biasplots(dsin=         /* required */
* ,  dsout=                      /* required */
* ,  title1="Test Title 1"       /* optional */
* ,  title2="Test Title 2"       /* optional */
* ,  title3="Test Title 3"       /* optional */
* ,  sampleid=                   /* required */
* ,  iter= 1                     /* optional */
* ,  xvar=                       /* required */
* ,  xlab= "Reference Label"     /* optional */
* ,  yvar=                       /* required */
* ,  ylab= "Test Label"          /* optional */
* ,  diffxy=                     /* optional */
* ,  diffxylab=                  /* optional */
* ,  meanxy=                     /* optional */
* ,  meanxylab=                  /* optional */
* ,  alpha= 0.05                 /* optional */
* ,  GraphMin=                   /* optional */
* ,  GraphMax=                   /* optional */
* ,  GraphIncrement= 1           /* optional */
* ,  BiasGraphMin=               /* optional */
* ,  BiasGraphMax=               /* optional */
* ,  BiasIncrement=              /* optional */
* ,  ParmDecimalPlaces= 0.001) ; /* optional */
*
*
* Macro Authors: Jesse A. Canchola (JAC), Natasha Oza (N-O)
* Creation Date: 12-AUG-2022
*
*
* Validator(s):  Natasha Oza (N-O)
* Validation Date: 12-AUG-2022
```

```
* ****************************************************************** ;
%macro biasplots(dsin=
,dsout=
,title1=
,title2=
,title3=
,sampleid=
,iter=
,xvar=
,xlab=
,yvar=
,ylab=
,diffxy=
,diffxylab=
,meanxy=
,meanxylab=
,alpha= 0.05
,GraphMin=
,GraphMax=
,GraphIncrement=
,BiasGraphMin     =
,BiasGraphMax     =
,BiasIncrement    =
,ParmDecimalPlaces= 0.001) ;

* Defaults Section ;
%if &iter.              = %then %let iter  = 1 ;
%if &alpha.             = %then %let alpha = 0.05 ;
%if &ParmDecimalPlaces. = %then %let ParmDecimalPlaces = 0.001 ;
%if &GraphIncrement.    = %then %let GraphIncrement = 1 ;
%if &BiasIncrement.     = %then %let BiasIncrement = 0.5 ;
ods escapechar='~' ;
* min and max of data for axis defaults other than SAS defaults
* if some or none of graphical parms are specified in macro ;
* variables created by proc sql are automatically macro variables
* when using "into" with a colon in front rather than "as" with no colon ;
proc sql noprint ;
  select min(&xvar.) into :xvar_min
  from &dsin. ;
quit;
proc sql noprint ;
  select max(&xvar.) into :xvar_max
  from &dsin. ;
quit;
proc sql noprint ;
  select min(&yvar.) into :yvar_min
  from &dsin. ;
quit;
proc sql noprint ;
  select max(&yvar.) into :yvar_max
  from &dsin. ;
quit;
%if &GraphMin. = %then %let GraphMin1 = &xvar_min. ;
%if &GraphMax. = %then %let GraphMax1 = &xvar_max. ;
%if &BiasGraphMin. = %then %let BiasGraphMin1 = &yvar_min. ;
%if &BiasGraphMax. = %then %let BiasGraphMax1 = &yvar_max. ;
* keep only complete valid pairs ;
proc sort data=&dsin. ; by &sampleid. ; run ;
data complete_tmp ;
  set &dsin. ;
  if &yvar. ne . or &xvar. = . ;

  * calculate y-x difference if not provided ;
  if %length(&diffxy.) = 0 then diffxy = &yvar. - &xvar. ;

  * calculate mean of x,y if not provided ;
  if %length(&meanxy.) = 0 then meanxy = mean(&xvar.,&yvar.) ;

  * calculating the log difference ;
  log10_yvar = log10(&yvar.) ;
  log10_xvar = log10(&xvar.) ;
```

```
  logdif = log10_yvar - log10_xvar ;
  logave = mean(log10_xvar,log10_yvar) ;

  * calculating the absolute value of the difference ;
  abs_logdif = abs(logdif) ;
run ;

* OLS regression ;
proc means data = complete_tmp n mean std min max maxdec=3 lclm uclm ;
%if &diffxy. = %then
        %do ;
          var diffxy ;
        %end ;
        %else %do ;
             var &diffxy. ;
        %end ;
  output out = complete_Mean_tmp
        mean = logdif_mean
        std  = logdif_stddev
        min  = logdif_min
        max  = logdif_max
        lclm = logdif_LCLM
        uclm = logdif_UCLM
        N    = logdif_N
      ;
run ;
* summary result calculations ;
data complete_Mean ;
 set complete_Mean_tmp (rename=
            (logdif_LCLM   = logdif_LCLM1
             logdif_UCLM   = logdif_UCLM1
             logdif_MAX    = logdif_MAX1
             logdif_MEAN   = logdif_MEAN1
             logdif_MIN    = logdif_MIN1
             logdif_N      = logdif_N1
             logdif_STDDEV = logdif_STDDEV1
             )
drop= _type_) ;*edit;

 logdif_lcl1 = logdif_mean1 - quantile("Normal", 1-&alpha./2) * logdif_stddev1  ;
 logdif_ucl1 = logdif_mean1 + quantile("Normal", 1-&alpha./2) * logdif_stddev1  ;

 CIP1 = (1-&alpha.)*100 ;
 CI1  = compress(CIP1||"%") ;
 call symputx("ci",CI1) ;
 %local ci ;
run ;
* individual result calculations ;
* merge outcome with mean data ;
data complete_tmp2 ;
 merge complete_tmp
       complete_Mean ;

 MatchVar = 1 ; *for retain below ;
run ;

* using RETAIN to backfill ;
proc sort data= complete_tmp2 ; by MatchVar ; run ;
data complete_tmp3 ;
  set complete_tmp2 ;
  by MatchVar ;
  retain
    logdif_lcl logdif_ucl logdif_N logdif_LCLM logdif_UCLM
    logdif_MAX logdif_MEAN logdif_MIN logdif_STDDEV ci cip;
  if first.MatchVar then
     do ;
        logdif_lcl    = logdif_lcl1     ;
        logdif_ucl    = logdif_ucl1     ;
        logdif_N      = logdif_N1       ;
        logdif_LCLM   = logdif_LCLM1    ;
        logdif_UCLM   = logdif_UCLM1    ;
```

```
        logdif_MAX    = logdif_MAX1     ;
        logdif_MEAN   = logdif_MEAN1    ;
        logdif_MIN    = logdif_MIN1     ;
        logdif_STDDEV = logdif_STDDEV1  ;
        ci            = ci1             ;
        cip           = cip1            ;
     end ;

  drop logdif_lcl1 logdif_ucl1 logdif_N1 logdif_LCLM1 logdif_UCLM1
       logdif_MAX1 logdif_MEAN1 logdif_MIN1 logdif_STDDEV1 ci1 cip1 MatchVar;
run ;

* summary result calculations ;
data complete       ;
 set complete_tmp3  ;

 * listing of values >= +-quantile("Normal", 1-alpha/2)SD units ;
 if logdif < logdif_lcl OR logdif > logdif_ucl then ObsOutside = 1 ;
 else ObsOutside = 0 ;
run ;

proc iml ;
use complete_Mean ;
read all VAR{logdif_mean1
             logdif_lclm1
             logdif_uclm1
             logdif_lcl1
             logdif_ucl1
             logdif_N1} into X ;
close complete_Mean ;

bias_i&iter. = X[1,1] ;
CI95_LL_bias_i&iter. = X[1,2] ;
CI95_UL_bias_i&iter. = X[1,3] ;

CI95I_LL_bias_i&iter. = X[1,4] ;
CI95I_UL_bias_i&iter. = X[1,5] ;

n_i&iter. = X[1,6] ;
call symputx("n_i&iter.",n_i&iter.) ;
call symputx("bias_i&iter.",round(bias_i&iter.,0.001)) ;
call symputx("LCI95_bias_i&iter.",round(CI95_LL_bias_i&iter.,0.001)) ;
call symputx("UCI95_bias_i&iter.",round(CI95_UL_bias_i&iter.,0.001)) ;
call symputx("LCI95I_bias_i&iter.",round(CI95I_LL_bias_i&iter.,0.001)) ;
call symputx("UCI95I_bias_i&iter.",round(CI95I_UL_bias_i&iter.,0.001)) ;

%local n_i&iter. bias_i&iter. LCI95_bias_i&iter. UCI95_bias_i&iter. LCI95I_bias_i&iter.
       UCI95I_bias_i&iter. ;
quit ;
run ;

* ************************** ;
* REGRESSION OF DIF BY AVE X,Y ;
* ************************** ;
ods output ParameterEstimates=Parm&iter. FitStatistics=FitStat&iter. NObs=NOb&iter. ;
proc reg data= complete ;
 model logdif = logave / clm cli clb ;
 output out=OLSresid&iter. predicted=pred5 residual=resid&iter. press=pres&iter. ;
run ; quit ;

proc iml ;
use Parm&iter. ;
read all VAR{estimate lowercl uppercl} into X ;
close Parm&iter. ;

use FitStat&iter. ;
read all VAR{NVALUE2} into Y ;
close FitStat&iter. ;

use NOb&iter. ;
read all VAR{N} into Z ;
```

```
close NOb&iter. ;

NOb&iter. = Z[1,1] ;

ols&iter._b0 = X[1,1] ;
ols&iter._b1 = X[2,1] ;

CI95_LL&iter._b0 = X[1,2] ;
CI95_UL&iter._b0 = X[1,3] ;

CI95_LL&iter._b1 = X[2,2] ;
CI95_UL&iter._b1 = X[2,3] ;

Rsquare&iter. = Y[1,1] ;

call symputx("N&iter.",Nob&iter.) ;

call symputx("ols&iter._b0",round(ols&iter._b0,0.001)) ;
call symputx("ols&iter._b1",round(ols&iter._b1,0.001)) ;

call symputx("CI95_LL&iter._b0",round(CI95_LL&iter._b0,0.001)) ;
call symputx("CI95_UL&iter._b0",round(CI95_UL&iter._b0,0.001)) ;

call symputx("CI95_LL&iter._b1",round(CI95_LL&iter._b1,0.001)) ;
call symputx("CI95_UL&iter._b1",round(CI95_UL&iter._b1,0.001)) ;

call symputx("Rsquare&iter.",round(Rsquare&iter.,0.001)) ;


* formatting a "clean" regression equation for sgplot ;
  abs_ols&iter._b1 = abs(ols&iter._b1) ;
  call symputx("abs_ols&iter._b1",round(abs_ols&iter._b1,0.001)) ;
  if ols&iter._b1 > 0 then RegEqn&iter. = "Y = &&ols&iter._b0 + &&ols&iter._b1 X" ;
  else RegEqn&iter. = "Y = &&ols&iter._b0 - &&abs_ols&iter._b1 X";
  call symputx("RegEqn&iter.",RegEqn&iter.) ;

%local NOb&iter. ols&iter._b0 ols&iter._b1 CI95_LL&iter._b0 CI95_UL&iter._b0
       CI95_LL&iter._b1 CI95_UL&iter._b1 Rsquare&iter. ;
quit ;
run ;

* ************************** ;
* REGRESSION OF DIF BY X       ;
* ************************** ;
ods output ParameterEstimates=ParmKW&iter. FitStatistics=FitStatKW&iter. NObs=NObKW&iter. ;
proc reg data= complete ;

 model logdif = &xvar. / clm cli clb ;


 output out=OLSresidKW&iter. predicted=pred5 residual=residKW&iter. press=presKW&iter. ;
run ; quit ;

proc iml ;
use ParmKW&iter. ;
read all VAR{estimate lowercl uppercl} into X ;
close ParmKW&iter. ;

use FitStatKW&iter. ;
read all VAR{NVALUE2} into Y ;
close FitStatKW&iter. ;

use NObKW&iter. ;
read all VAR{N} into Z ;
close NObKW&iter. ;

NObKW&iter. = Z[1,1] ;

olsKW&iter._b0 = X[1,1] ;
olsKW&iter._b1 = X[2,1] ;
```

```
CI95_LLKW&iter._b0 = X[1,2] ;
CI95_ULKW&iter._b0 = X[1,3] ;

CI95_LLKW&iter._b1 = X[2,2] ;
CI95_ULKW&iter._b1 = X[2,3] ;

RsquareKW&iter. = Y[1,1] ;

call symputx("NKW&iter.",NobKW&iter.) ;

call symputx("olsKW&iter._b0",round(olsKW&iter._b0,0.001)) ;
call symputx("olsKW&iter._b1",round(olsKW&iter._b1,0.001)) ;

call symputx("CI95_LLKW&iter._b0",round(CI95_LLKW&iter._b0,0.001)) ;
call symputx("CI95_ULKW&iter._b0",round(CI95_ULKW&iter._b0,0.001)) ;

call symputx("CI95_LLKW&iter._b1",round(CI95_LLKW&iter._b1,0.001)) ;
call symputx("CI95_ULKW&iter._b1",round(CI95_ULKW&iter._b1,0.001)) ;

call symputx("RsquareKW&iter.",round(RsquareKW&iter.,0.001)) ;


* formatting a "clean" regression equation for sgplot ;
  abs_olsKW&iter._b1 = abs(olsKW&iter._b1) ;
  call symputx("abs_olsKW&iter._b1",round(abs_olsKW&iter._b1,0.001)) ;
  if olsKW&iter._b1 > 0 then RegEqnKW&iter. = "Y = &&olsKW&iter._b0 + &&olsKW&iter._b1 X" ;
  else RegEqnKW&iter. = "Y = &&olsKW&iter._b0 - &&abs_olsKW&iter._b1 X";
  call symputx("RegEqnKW&iter.",RegEqnKW&iter.) ;

%local NObKW&iter. olsKW&iter._b0 olsKW&iter._b1 CI95_LLKW&iter._b0 CI95_ULKW&iter._b0
       CI95_LLKW&iter._b1 CI95_ULKW&iter._b1 RsquareKW&iter. ;
quit ;
run ;

******************** BEGIN BIAS PLOTS ******************** ;

************************ ;
* BLAND-ALTMAN BIAS PLOTS ;
************************ ;
footnote1 " " ;
footnote2 "Bland-Altman Bias Plot" ;
proc sgplot data = complete noborder ;
* diffxy and/or meanxy not entered ;
%if &diffxy. = and &meanxy. = %then
        %do ;
              scatter y= diffxy   x= meanxy / MARKERATTRS  = (size=8px) name = "Scatter" ;
        %end ;
%else %if &diffxy. = and &meanxy. ne %then
        %do ;
              scatter y= diffxy   x= &meanxy. / MARKERATTRS  = (size=8px) name = "Scatter" ;
        %end ;
%else %if &diffxy. ne and &meanxy. = %then
        %do ;
              scatter y= &diffxy. x= meanxy / MARKERATTRS  = (size=8px) name = "Scatter" ;
        %end ;
%else %if &diffxy. ne and &meanxy. ne %then
        %do ;
              scatter y= &diffxy. x= &meanxy. / MARKERATTRS  = (size=8px) name = "Scatter" ;
        %end ;

  REFLINE &&bias_i&iter.        / TRANSPARENCY = 0.5
        LINEATTRS = (THICKNESS=0.1 color=Black PATTERN=DOT)
        LABEL = ("Bias")    LABELATTRS=(size=8 color=black family=Imago ) ;
  REFLINE &&LCI95I_bias_i&iter. / TRANSPARENCY = 0.5
        LINEATTRS = (THICKNESS=0.1 color=Black PATTERN=DOT)
        LABEL = ("&ci. LL") LABELATTRS=(size=8 color=black family=Imago ) ;
  REFLINE &&UCI95I_bias_i&iter. / TRANSPARENCY = 0.5
        LINEATTRS = (THICKNESS=0.1 color=Black PATTERN=DOT)
        LABEL = ("&ci. UL") LABELATTRS=(size=8 color=black family=Imago ) ;

%if &BiasGraphMin. eq and &BiasGraphMax. eq
```

```
        %then %do ;
                YAXIS LABEL = &diffxylab. ;
        %end ;
        %else %do ;
                YAXIS LABEL = &diffxylab.
                  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 = &meanxylab. ;
          %end ;
        %else %do ;
                XAXIS LABEL = &meanxylab.
                  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 "Mean difference (N=&&N&iter.): &&bias_i&iter. log10" "&ci. CI: (&&LCI95_bias_i&iter.,
&&UCI95_bias_i&iter.)"
          / POSITION = BOTTOMRIGHT NOBORDER TEXTATTRS=(color=Black family=Imago size=8px) ;

  *KEYLEGEND "Scatter" / LOCATION = outside POSITION = bottom ;

  TITLE1 &title1. ;
  TITLE2 &title2. ;
  TITLE3 &title3. ;
run ;

proc sgplot data = complete noborder ;
* diffxy and/or meanxy not entered ;
%if &diffxy. = and &meanxy. = %then
          %do ;
             scatter y= diffxy   x= meanxy / MARKERATTRS  = (size=8px) name = "Scatter" ;
          %end ;
%else %if &diffxy. = and &meanxy. ne %then
          %do ;
             scatter y= diffxy   x= &meanxy. / MARKERATTRS  = (size=8px) name = "Scatter" ;
```

```
          %end ;
%else %if &diffxy. ne and &meanxy. = %then
          %do ;
             scatter y= &diffxy. x= meanxy / MARKERATTRS  = (size=8px) name = "Scatter" ;
          %end ;
%else %if &diffxy. ne and &meanxy. ne %then
          %do ;
             scatter y= &diffxy. x= &meanxy. / MARKERATTRS  = (size=8px) name = "Scatter" ;
          %end ;

  REFLINE &&bias_i&iter.        / TRANSPARENCY = 0.5
           LINEATTRS = (THICKNESS=0.1 color=Black PATTERN=DOT) LABEL = ("Bias=&&bias_i&iter.")
           LABELATTRS=(size=8 color=black family=Imago ) ;
  REFLINE &&LCI95I_bias_i&iter. / TRANSPARENCY = 0.5
           LINEATTRS = (THICKNESS=0.1 color=Black PATTERN=DOT)
           LABEL = ("&ci. LL=&&LCI95I_bias_i&iter.")
           LABELATTRS=(size=8 color=black family=Imago ) ;
  REFLINE &&UCI95I_bias_i&iter. / TRANSPARENCY = 0.5
           LINEATTRS = (THICKNESS=0.1 color=Black PATTERN=DOT)
           LABEL = ("&ci. UL=&&UCI95I_bias_i&iter.")
           LABELATTRS=(size=8 color=black family=Imago ) ;

%if &BiasGraphMin. eq and &BiasGraphMax. eq
    %then %do ;
                YAXIS LABEL = &diffxylab. ;
          %end ;
       %else %do ;
                YAXIS LABEL = &diffxylab.
                  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 = &meanxylab. ;
          %end ;
       %else %do ;
                XAXIS LABEL = &meanxylab.
                  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 "Mean difference (N=&&N&iter.): &&bias_i&iter. log10" "&ci. CI: (&&LCI95_bias_i&iter.,
&&UCI95_bias_i&iter.)"
          / POSITION = BOTTOMRIGHT NOBORDER TEXTATTRS=(color=Black family=Imago size=8px) ;

  *KEYLEGEND "Scatter" / LOCATION = outside POSITION = bottom ;

  TITLE1 &title1. ;
  TITLE2 &title2. ;
  TITLE3 &title3. ;
run ;


proc sgplot data = complete noborder ;
* diffxy and/or meanxy not entered ;
%if &diffxy. = and &meanxy. = %then
          %do ;
             reg y= diffxy   x= meanxy / MARKERATTRS  = (size=8px) name = "Regression" ;
          %end ;
%else %if &diffxy. = and &meanxy. ne %then
          %do ;
             reg y= diffxy   x= &meanxy. / MARKERATTRS  = (size=8px) name = "Regression" ;
          %end ;
%else %if &diffxy. ne and &meanxy. = %then
          %do ;
             reg y= &diffxy. x= meanxy / MARKERATTRS  = (size=8px) name = "Regression" ;
          %end ;
%else %if &diffxy. ne and &meanxy. ne %then
          %do ;
             reg y= &diffxy. x= &meanxy. / MARKERATTRS  = (size=8px) name = "Regression" ;
          %end ;

  REFLINE &&bias_i&iter.        / TRANSPARENCY = 0.5
          LINEATTRS = (THICKNESS=0.1 color=Black PATTERN=DOT)
          LABEL = ("Bias=&&bias_i&iter.")
          LABELATTRS=(size=8 color=black family=Imago ) ;
  REFLINE &&LCI95I_bias_i&iter. / TRANSPARENCY = 0.5
          LINEATTRS = (THICKNESS=0.1 color=Black PATTERN=DOT)
          LABEL = ("&ci. LL=&&LCI95I_bias_i&iter.")
          LABELATTRS=(size=8 color=black family=Imago ) ;
  REFLINE &&UCI95I_bias_i&iter. / TRANSPARENCY = 0.5
          LINEATTRS = (THICKNESS=0.1 color=Black PATTERN=DOT)
          LABEL = ("&ci. UL=&&UCI95I_bias_i&iter.")
          LABELATTRS=(size=8 color=black family=Imago ) ;

%if &BiasGraphMin. eq and &BiasGraphMax. eq
    %then %do ;
                YAXIS LABEL = &diffxylab. ;
          %end ;
       %else %do ;
                YAXIS LABEL = &diffxylab.
                  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 = &meanxylab. ;
          %end ;
       %else %do ;
                XAXIS LABEL = &meanxylab.
                  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 "OLS Regression (N=&&N&iter.): &&RegEqn&iter."
        "95% CI Slope: (&&CI95_LL&iter._b1, &&CI95_UL&iter._b1)"
        / POSITION = BOTTOMRIGHT NOBORDER TEXTATTRS=(color=Black family=Imago size=8px) ;

  *KEYLEGEND "Regression" / LOCATION = outside POSITION = bottom ;

  TITLE1 &title1. ;
  TITLE2 &title2. ;
  TITLE3 &title3. ;
run ;

proc sgplot data = complete ;
* diffxy and/or meanxy not entered ;
%if &diffxy. = and &meanxy. = %then
         %do ;
             scatter y= diffxy   x= meanxy / MARKERATTRS  = (size=8px) name = "Scatter" ;
          %end ;
%else %if &diffxy. = and &meanxy. ne %then
          %do ;
             scatter y= diffxy   x= &meanxy. / MARKERATTRS  = (size=8px) name = "Scatter" ;
          %end ;
%else %if &diffxy. ne and &meanxy. = %then
         %do ;
             scatter y= &diffxy. x= meanxy / MARKERATTRS  = (size=8px) name = "Scatter" ;
          %end ;
%else %if &diffxy. ne and &meanxy. ne %then
         %do ;
             scatter y= &diffxy. x= &meanxy. / MARKERATTRS  = (size=8px) name = "Scatter" ;
          %end ;

  REFLINE &&bias_i&iter.        / TRANSPARENCY = 0.5
          LINEATTRS = (THICKNESS=0.1 COLOR=black PATTERN=DOT)    ;
  REFLINE &&LCI95I_bias_i&iter. / TRANSPARENCY = 0.5
          LINEATTRS = (THICKNESS=0.1 COLOR=black PATTERN=dash) ;
  REFLINE &&UCI95I_bias_i&iter. / TRANSPARENCY = 0.5
          LINEATTRS = (THICKNESS=0.1 COLOR=black PATTERN=dash) ;

%if &BiasGraphMin. eq and &BiasGraphMax. eq
    %then %do ;
                YAXIS LABEL = &diffxylab. ;
          %end ;
    %else %do ;
                YAXIS LABEL = &diffxylab.
```

```
                  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 = &meanxylab. ;
          %end ;
        %else %do ;
                XAXIS LABEL = &meanxylab.
                  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 ;

  TITLE1 ;
  TITLE2 ;
  TITLE3 ;
run ;
footnote1 ; footnote2 ;

************************ ;
* Krouwer BIAS PLOTS      ;
************************ ;
footnote1 " " ;
footnote2 "Krouwer Bias Plot" ;
proc sgplot data = complete noborder ;
* diffxy and/or &xvar. not entered ;
%if &diffxy. = and &xvar. = %then
         %do ;
             scatter y= diffxy   x= &xvar. / MARKERATTRS  = (size=8px) name = "Scatter" ;
          %end ;
%else %if &diffxy. = and &xvar. ne %then
         %do ;
             scatter y= diffxy   x= &xvar. / MARKERATTRS  = (size=8px) name = "Scatter" ;
          %end ;
%else %if &diffxy. ne and &xvar. = %then
         %do ;
             scatter y= &diffxy. x= &xvar. / MARKERATTRS  = (size=8px) name = "Scatter" ;
          %end ;
```

```
%else %if &diffxy. ne and &xvar. ne %then
          %do ;
             scatter y= &diffxy. x= &xvar. / MARKERATTRS  = (size=8px) name = "Scatter" ;
           %end ;

  REFLINE &&bias_i&iter.         / TRANSPARENCY = 0.5
          LINEATTRS = (THICKNESS=0.1 color=Black PATTERN=DOT) LABEL = ("Bias")
          LABELATTRS=(size=8 color=black family=Imago ) ;
  REFLINE &&LCI95I_bias_i&iter. / TRANSPARENCY = 0.5
          LINEATTRS = (THICKNESS=0.1 color=Black PATTERN=DOT)
          LABEL = ("&ci. LL") LABELATTRS=(size=8 color=black family=Imago ) ;
  REFLINE &&UCI95I_bias_i&iter. / TRANSPARENCY = 0.5
          LINEATTRS = (THICKNESS=0.1 color=Black PATTERN=DOT)
          LABEL = ("&ci. UL")
          LABELATTRS=(size=8 color=black family=Imago ) ;

%if &BiasGraphMin. eq and &BiasGraphMax. eq
    %then %do ;
                YAXIS LABEL = &diffxylab. ;
          %end ;
       %else %do ;
                YAXIS LABEL = &diffxylab.
                  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 = &xlab. ;
          %end ;
       %else %do ;
                XAXIS LABEL = &xlab.
                  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 "Mean difference (N=&&NKW&iter.): &&bias_i&iter. log10"
        "&ci. CI: (&&LCI95_bias_i&iter., &&UCI95_bias_i&iter.)"
          / POSITION = BOTTOMRIGHT NOBORDER TEXTATTRS=(color=Black family=Imago size=8px) ;
```

```
  *KEYLEGEND "Scatter" / LOCATION = outside POSITION = bottom ;

  TITLE1 &title1. ;
  TITLE2 &title2. ;
  TITLE3 &title3. ;
run ;

proc sgplot data = complete noborder ;
* diffxy and/or &xvar. not entered ;
%if &diffxy. = and &xvar. = %then
         %do ;
             scatter y= diffxy   x= &xvar. / MARKERATTRS  = (size=8px) name = "Scatter" ;
          %end ;
%else %if &diffxy. = and &xvar. ne %then
         %do ;
             scatter y= diffxy   x= &xvar. / MARKERATTRS  = (size=8px) name = "Scatter" ;
          %end ;
%else %if &diffxy. ne and &xvar. = %then
         %do ;
             scatter y= &diffxy. x= &xvar. / MARKERATTRS  = (size=8px) name = "Scatter" ;
          %end ;
%else %if &diffxy. ne and &xvar. ne %then
         %do ;
             scatter y= &diffxy. x= &xvar. / MARKERATTRS  = (size=8px) name = "Scatter" ;
          %end ;

  REFLINE &&bias_i&iter.        / TRANSPARENCY = 0.5
          LINEATTRS = (THICKNESS=0.1 color=Black PATTERN=DOT)
          LABEL = ("Bias=&&bias_i&iter.")
          LABELATTRS=(size=8 color=black family=Imago ) ;
  REFLINE &&LCI95I_bias_i&iter. / TRANSPARENCY = 0.5
          LINEATTRS = (THICKNESS=0.1 color=Black PATTERN=DOT)
          LABEL = ("&ci. LL=&&LCI95I_bias_i&iter.")
          LABELATTRS=(size=8 color=black family=Imago ) ;
  REFLINE &&UCI95I_bias_i&iter. / TRANSPARENCY = 0.5
          LINEATTRS = (THICKNESS=0.1 color=Black PATTERN=DOT)
          LABEL = ("&ci. UL=&&UCI95I_bias_i&iter.")
          LABELATTRS=(size=8 color=black family=Imago ) ;

%if &BiasGraphMin. eq and &BiasGraphMax. eq
    %then %do ;
                YAXIS LABEL = &diffxylab. ;
          %end ;
       %else %do ;
                YAXIS LABEL = &diffxylab.
                  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 = &xlab. ;
          %end ;
       %else %do ;
```

```
                XAXIS LABEL = &xlab.
                  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 "Mean difference (N=&&NKW&iter.): &&bias_i&iter. log10"
        "&ci. CI: (&&LCI95_bias_i&iter., &&UCI95_bias_i&iter.)"
          / POSITION = BOTTOMRIGHT NOBORDER TEXTATTRS=(color=Black family=Imago size=8px) ;

  *KEYLEGEND "Scatter" / LOCATION = outside POSITION = bottom ;

  TITLE1 &title1. ;
  TITLE2 &title2. ;
  TITLE3 &title3. ;
run ;


proc sgplot data = complete noborder ;
* diffxy and/or &xvar. not entered ;
%if &diffxy. = and &xvar. = %then
          %do ;
              reg y= diffxy   x= &xvar. / MARKERATTRS  = (size=8px) name = "Regression" ;
           %end ;
%else %if &diffxy. = and &xvar. ne %then
          %do ;
              reg y= diffxy   x= &xvar. / MARKERATTRS  = (size=8px) name = "Regression" ;
           %end ;
%else %if &diffxy. ne and &xvar. = %then
          %do ;
              reg y= &diffxy. x= &xvar. / MARKERATTRS  = (size=8px) name = "Regression" ;
           %end ;
%else %if &diffxy. ne and &xvar. ne %then
          %do ;
              reg y= &diffxy. x= &xvar. / MARKERATTRS  = (size=8px) name = "Regression" ;
           %end ;

  REFLINE &&bias_i&iter.        / TRANSPARENCY = 0.5
           LINEATTRS = (THICKNESS=0.1 color=Black PATTERN=DOT)
           LABEL = ("Bias=&&bias_i&iter.")
           LABELATTRS=(size=8 color=black family=Imago ) ;
  REFLINE &&LCI95I_bias_i&iter. / TRANSPARENCY = 0.5
           LINEATTRS = (THICKNESS=0.1 color=Black PATTERN=DOT)
           LABEL = ("&ci. LL=&&LCI95I_bias_i&iter.")
           LABELATTRS=(size=8 color=black family=Imago ) ;
  REFLINE &&UCI95I_bias_i&iter. / TRANSPARENCY = 0.5
           LINEATTRS = (THICKNESS=0.1 color=Black PATTERN=DOT)
           LABEL = ("&ci. UL=&&UCI95I_bias_i&iter.")
           LABELATTRS=(size=8 color=black family=Imago ) ;

%if &BiasGraphMin. eq and &BiasGraphMax. eq
    %then %do ;
              YAXIS LABEL = &diffxylab. ;
          %end ;
       %else %do ;
              YAXIS LABEL = &diffxylab.
```

```
                  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 = &xlab. ;
          %end ;
        %else %do ;
                XAXIS LABEL = &xlab.
                  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 "OLS Regression (N=&&NKW&iter.): &&RegEqnKW&iter."
        "95% CI Slope: (&&CI95_LLKW&iter._b1, &&CI95_ULKW&iter._b1)"
        / POSITION = BOTTOMRIGHT NOBORDER TEXTATTRS=(color=Black family=Imago size=8px) ;

  TITLE1 &title1. ;
  TITLE2 &title2. ;
  TITLE3 &title3. ;
run ;

proc sgplot data = complete ;
* diffxy and/or &xvar. not entered ;
%if &diffxy. = and &xvar. = %then
          %do ;
              scatter y= diffxy   x= &xvar. / MARKERATTRS  = (size=8px) name = "Scatter" ;
           %end ;
%else %if &diffxy. = and &xvar. ne %then
          %do ;
              scatter y= diffxy   x= &xvar. / MARKERATTRS  = (size=8px) name = "Scatter" ;
           %end ;
%else %if &diffxy. ne and &xvar. = %then
          %do ;
              scatter y= &diffxy. x= &xvar. / MARKERATTRS  = (size=8px) name = "Scatter" ;
           %end ;
%else %if &diffxy. ne and &xvar. ne %then
```

```
          %do ;
              scatter y= &diffxy. x= &xvar. / MARKERATTRS  = (size=8px) name = "Scatter" ;
           %end ;

  REFLINE &&bias_i&iter.        / TRANSPARENCY = 0.5
          LINEATTRS = (THICKNESS=0.1 COLOR=black PATTERN=DOT)    ;
  REFLINE &&LCI95I_bias_i&iter. / TRANSPARENCY = 0.5
          LINEATTRS = (THICKNESS=0.1 COLOR=black PATTERN=dash) ;
  REFLINE &&UCI95I_bias_i&iter. / TRANSPARENCY = 0.5
          LINEATTRS = (THICKNESS=0.1 COLOR=black PATTERN=dash) ;

%if &BiasGraphMin. eq and &BiasGraphMax. eq
    %then %do ;
                YAXIS LABEL = &diffxylab. ;
          %end ;
        %else %do ;
                YAXIS LABEL = &diffxylab.
                  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 = &xlab. ;
          %end ;
        %else %do ;
                XAXIS LABEL = &xlab.
                  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 ;

  TITLE1 ;
  TITLE2 ;
  TITLE3 ;
run ;
footnote1 ; footnote2 ;
%MEND biasplots ;
```

APPENDIX B: BIASPLOTS SAS MACRO CALL FOR EXAMPLE 1

```
%LET SEED1 = 726453819 ;
%LET SEED2 = 289546394 ;
%LET SEED3 = 923847362 ;
ods listing close;
data Example ;
 do MyId = 1 to 200 ;
  Value = UNIFORM(&SEED1.) * 10 ;
  x1 = abs(Value + NORMAL(&SEED2.)) ;
  y1 = abs(Value + NORMAL(&SEED3.)) ;
  yx_diff= y1-x1;
  yx_avg=(y1+x1)/2;
  log10_yvar = log10(y1) ;
  log10_xvar = log10(x1) ;
  logdif = log10_yvar - log10_xvar ;
  logave = mean(log10_xvar,log10_yvar) ;
  output ;
 end ;
 drop Value ;
run ;
%macro special_char(unicode=, name=);
 %global &name;
 data _null_;
 A=input("&unicode."x,$UCS2B4.);
 call symput("&name.",trim(left(A)));
 stop;
 run;
 %put Note: special_char: &name = ->|&&&name.|<- ;
%mend;
%special_char(unicode=00AE,name=rball) ;
* Example: title "Title with Registered Trademark &rball." ;

* Use the MODSTYLE macro supplied by SAS to cange the default colors and markers ;
%modstyle(name    = markers
         ,parent  = listing
         ,type    = CLM
         ,markers = circle triangle square diamond star circlefilled trianglefilled diamondfilled
         ,colors  = blue green red purple orange black pink brown
         ,linestyles = Solid LongDash MediumDashDotDot  ShortDashDot ThinDot ShortDash Dot
                       MediumDash) ;

* The investigators want a square plot (no proc template necessary) ;
ods _all_ close ;
ods listing;
ods graphics on;
ods listing gpath="c:/sasmacros/example/output" ;
ods escapechar='~' ;
ods graphics / width = 550px height = 550px noborder outputfmt=PNG ; *SVG ; *EMF ; *tiff ;
title1 "Bias Plots" ;

%include "c:/sasmacros/biasplots.sas" ;
ods noproctitle ;
ods rtf file = "c:/sasmacros/example/output/out1.rtf" gtitle ;
%biasplots(dsin=example
,dsout=example_Out
,title1="title1"
,title2="title2"
,title3=
,sampleid=myid
,iter= 1
,xvar= x1
,xlab= "x-axis"
,yvar= y1
,ylab= "y-axis"
,diffxy=
,diffxylab= "difference label"
,meanxy=
,meanxylab= "avg label"
,alpha= 0.05
,GraphMin=
```

```
,GraphMax=
,GraphIncrement=
,BiasGraphMin     =
,BiasGraphMax     =
,BiasIncrement    =
,ParmDecimalPlaces= 0.001) ;
ods rtf close ;

********************** end of program ********************** ;
```

□