Colorado School of PUBLIC HEALTH

UF Health Outcomes & Policy College of Medicine

Bias Correction Suite User Manual

Version 1.0.0

Authors:
Aarti Munjal and Brandy Ringham

April 2013

 Bias Correction Suite Version 1.0.0 is funded by NIDCR 1 R01 DE020832-01A1S, a grant awarded to the University of Florida (Keith E. Muller, PI; Deborah Glueck, University of Colorado site PI). Previous funding was received from NCI 1R03CA136048-01A1, a grant awarded to the Colorado School of Public Health, Deborah Glueck, Principal Investigator..

Contents

1. Introduction

1.1. Version information and licensing

This manual describes Bias Correction Suite version 1.0.0.

The Bias Correction Suite is released under the GNU Public License version 2.0.

Users can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, using either version 2 of the License, or any later version. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY—without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If you have not received a copy of the GNU General Public License and would like one, please write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

1.2. Welcome to Bias Correction Suite 1.0.0

Bias Correction Suite 1.0.0 is an open-source software. The software is a collection of SAS macros and SAS/IML modules. The software implements the method of Ringham et al. (in review). The method corrects for paired screening trial bias, a bias which can occur in studies using a paired comparison of the areas under the receiver operating characteristic curves (AUCs) of continuous screening tests (Glueck, Lamb, O'Donnell, Ringham, Brinton, Muller, Lewin, Alonzo, and Pisano 2009). Ringham et al. use a weighted maximum-likelihood approach to reduce decision errors in the paired comparison of cancer screening tests with Gaussian outcomes.

The Bias Correction Suite contains two main modules, the BiasCorrectionTool and the DecisionErrorTool. Scientists can use the two modules to 1) determine if bias correction is needed for their study (DecisionErrorTool), and 2) correct the bias in the results of their study (BiasCorrectionTool). The modules use the methods of Obuchowski and McClish (1997) to conduct a Z test of the difference in AUCs of two screening tests with Gaussian outcomes. The hypothesis test is conducted for two different analyses, the standard analysis and the bias-corrected analysis. The standard analysis uses the sample means, variances, and correlations of the screening test scores to calculate the test statistics. The bias-corrected analysis uses the bias-corrected means, variances, and correlations, as described in Ringham et al. (in review). With the BiasCorrectionTool, scientists input a dataset of screening test outcomes and obtain results for the standard and bias-corrected analyses. With the DecisionErrorTool, scientists simulate data from a paired screening trial. The simulation uses inputs that match what the scientist expects to observe in the planned study. Using the simulated data, the DecisionErrorTool presents the results of the standard and bias-corrected analyses and a third, reference analysis. The reference analysis assumes the scientist knows the true disease status of each participant. In reality, paired screening trials can be biased because the scientist does not observe the true disease status for every participant. Thus, the reference analysis uses unobserved information and is presented only as a point of reference to quantify the amount of error in the other two analyses. The DecisionErrorTool calculates the Type I error rate and power for all three analyses. Scientists can then choose an analysis plan (either standard or bias-corrected) that has the highest power for the correct decision and a nominal Type I error rate. We demonstrate the utility of the software with an application to a hypothetical oral cancer screening study.

2. When to use the Bias Correction Suite

The Bias Correction Suite is a tool for use by researchers to reduce decision errors in paired cancer screening trials.

We recommend our software for scientists using a paired screening trial with the following characteristics:

1. The goal of the paired cancer screening trial is to determine which of two screening tests has the best diagnostic accuracy.

2. The primary outcome of interest is the difference in the AUCs.

3. Study participants are screened with both screening tests.

4. Screening test scores are assumed to follow a bivariate Gaussian distribution conditional on the disease status.

5. Participants who screen negative on both tests enter a follow-up period.

6. Participants who screen positive on at least one screening test, or who show signs and symptoms of disease during the follow-up period, are given a gold standard test to determine their disease status.

7. The results of the gold standard test are assumed to be 100% sensitive and specific.

8. Participants who screen negative on both tests and who do not show signs and symptoms of disease during the follow-up period are assumed to be disease free.

3. How to set up the Bias Correction Suite

We outline the steps required to set up and use the Bias Correction Suite below:

1. Download the current release of Bias Correction Suite from www.samplesizeshop.org. Save the file on your computer, maintaining the internal file structure.

2. The Bias Correction Suite utilizes stored modules. To store the modules, run the *StoreBiasCorrectionModules* batch file included in the "Modules" folder by double-clicking on the filename. The program will compile and store all required modules in the "Library" folder.

3. If you will be using the BiasCorrectionTool, save your dataset in the "Input" folder. See Section 4.5.3 for a description of how to format your dataset.

4. How to use the Bias Correction Suite

4.1. Overview

The Bias Correction Suite is implemented as two separate SAS macros: BiasCorrectionTool and DecisionErrorTool. The suite also contains a set of stored function modules used by the two main SAS macros. The software relies on relative pathnames to access stored function modules, link to input datasets, and save output datasets and reports. The relative pathnames utilize a rigid file structure, provided in the Bias Correction Suite download. Section 4.2 describes the organization and purpose of the included files, Section 4.4 describes how to include the stored modules, and Sections 4.5 and 4.6 describe how to use the BiasCorrectionTool and DecisionErrorTool.

4.2. Bias Correction Suite file structure

The Bias Correction Suite file contains the file structure needed to implement the BiasCorrectionTool and DecisionErrorTool modules. Figure 1 illustrates the organization of the included folders. We describe each of the folders below:

Input: The "Input" folder contains the dataset used for the example in Section 5. User input datasets should be stored in this folder to ensure that the BiasCorrectionTool will successfully import the dataset.

Output: The "Output" folder contains the SAS datasets and summary reports output by the BiasCorrectionTool

or DecisionErrorTool. The "Output" folder includes two, optional, sub-folders for organizing output. Section 4.4 describes how to specify the organization of the SAS output.

Programs: The "Programs" folder contains all SAS programs used to implement the Bias Correction Suite, including both modules and main programs.

Modules: The "Modules" folder contains all modules used to implement the Bias Correction Suite, including the BiasCorrectionTool and DecisionErrorTool macros.

Main: The "Main" folder contains two example programs that call the BiasCorrectionTool and DecisionErrorTool. User created main programs should also be stored and run from this folder. This will ensure that the `%include` and `libname` statements point to the correct relative folder locations.

Library: The "Library" folder contains stored, compiled modules. See Section 4.4 for a description of how to store modules.

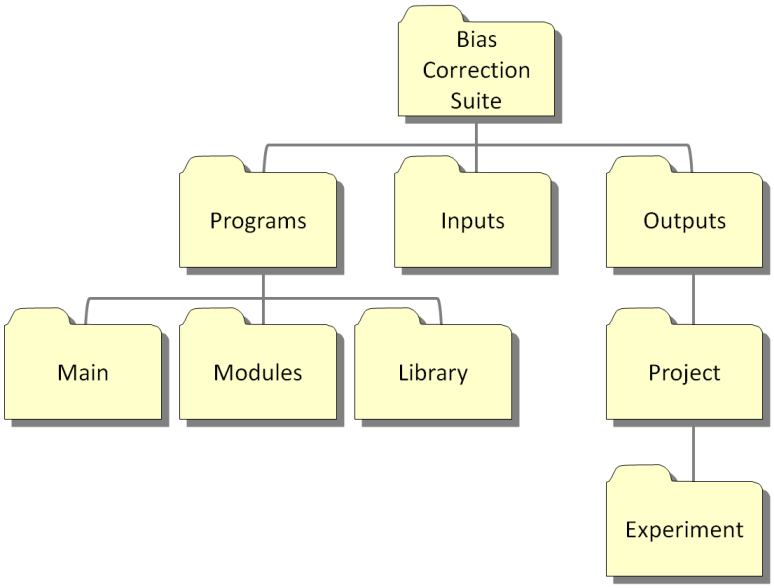

Figure 1: Bias Correction Suite required file structure.

4.3. How to start and run a new program

To start a new program, open up the appropriate example SAS program in the "Main" folder. Save the program as a new program. Type in comments and arguments relevant to your study. We describe all required arguments in the sections below. Press "Run" in the SAS window. Throughout, we refer to the program that invokes the BiasCorrectionTool or DecisionErrorTool as the *main* program.

4.4. How to specify the location of stored modules and output files

The BiasCorrectionTool and DecisionErrorTool utilize stored modules. To include the stored modules and specify where output should be stored, begin the main program with the following syntax:

```
%include "..\Modules\IncludeLibrary.sas";

%IncludeLibrary;
```

```
%CommonMain( <project>, <experiment>, <programName> );
```

The arguments `<project>` and `<experiment>` are optional. They allow the user to subdivide output into up to two sub-folders that take on the labels given to the arguments. See Figure 1 for an illustration of the file organization. The `<programName>` argument is the name the user would like to give to the output files. We typically define the `<programName>` argument as the name of the main program in order to track which main program created which output.

4.5. How to use the BiasCorrectionTool

4.5.1. Overview

The BiasCorrectionTool module corrects the bias in the difference in the AUCs of two screening tests with Gaussian outcomes. The module takes a user dataset and a set of biopsy thresholds as inputs. The user dataset contains the paired screening test scores for each participant and an indicator of the disease status (case or non-case) of each participant. The biopsy thresholds are the screening test specific thresholds above which screening test scores are considered positive and below which screening test scores are considered negative. The module conducts the standard analysis and the bias-corrected analysis on the user dataset.

4.5.2. The BiasCorrectionTool algorithm

Algorithm 1 outlines the steps in the BiasCorrectionTool.

Algorithm 1 BiasCorrectionTool

1.1 Partition participants into observed cases and non-cases.
1.2 Calculate the sample mean, variance, and correlation for the screening test scores of the observed cases and non-cases.
1.3 Calculate the difference in the areas under the binormal receiver operating characteristic curves (pg. 122, Zhou, McClish, and Obuchowski 2002) based on the sample means, variances, and correlations of the observed case and non-case screening test scores.
1.4 Calculate a z-statistic for the standard analysis, which uses the sample means, variances, and correlations as inputs.
1.5 Determine whether the hypothesis test for the standard analysis accepts or rejects.
1.6 Further partition participants into four quadrants: participants with screening test scores above both biopsy thresholds, participants with screening test scores below one threshold but above the other, and participants with screening test scores below both biopsy thresholds.
1.7 Obtain the maximum likelihood estimates (MLEs) of the bivariate Gaussian parameters for each quadrant of observed case screening test scores. The algorithm uses the method of Nath (1971) to calculate MLEs from truncated regions of the bivariate Gaussian distribution of screening test scores.
1.8 Calculate weighted parameter estimates from the quadrant specific MLEs to lower the variance.
1.9 Use the weighted parameter estimates to calculate the bias-corrected means, variances, and correlation of the observed case screening test scores.
1.10 Calculate the bias-corrected difference in the areas under the binormal receiver operating characteristic curves (pg. 122, Zhou *et al.* 2002) based on the bias-corrected means, variances, and correlations of the observed case screening test scores and the sample means, variances, and correlations of the observed non-case screening test scores.
1.11 Calculate a z-statistic for the bias-corrected analysis, which uses the bias-corrected difference in the AUCs.
1.12 Determine whether the hypothesis test for the bias-corrected analysis accepts or rejects.
1.13 Generate a report.

To perform the standard and bias corrected analyses, the BiasCorrectionTool contains five predefined submodules, as outlined below:

- *nathInAllQuadrants,* which calculates the MLEs for the screening test scores of the observed cases in each quadrant. The module first checks that the quadrants contain two or more observations (Steps 1.6-1.7).
- *standardAnalysis,* which performs the standard analysis (Steps 1.2-1.5).
- *biasCorrectionAnalysis,* which performs the bias-corrected analysis (*nathInAllQuadrants* and Steps 1.8-1.12).
- *setErrorCodes,* which sets an error code for variables based on the error encountered at several steps in the *biasCorrectionAnalysis* submodule.
- *performBiasCorrect,* which reads and analyzes the dataset entered by the user (Step 1.1, *standardAnalysis*, and *biasCorrectionAnalysis*).

In turn, the submodules call a series of stored modules, which are listed below.

- *calcCorrectedNumCasesNonCases,* which calculates the corrected number of cases and non-cases based on estimated bivariate normal probabilities.
- *calculateAUC,* which calculates the AUC assuming it is distributed binormal, using the method described in Zhou *et al.* (2002), pgs. 122, 139.

- *calculateWeightedEstimates,* which calculates weighted estimates of the parameters of the bivariate normal using estimates of the bivariate normal probabilities of each quadrant.
- *DifferenceInAUCHypothesisTest,* which conducts a hypothesis test for the difference in AUC using the methods of Obuchowski and McClish (1997).
- *estimateBivNormProbabilities*, which calculates the bivariate normal probabilities from estimated means, variances, and correlations.
- *maximizeLogLikelihood,* which maximizes the log likelihood of the bivariate normal distribution.
- *nathAlgorithm,* which implements the Nath algorithm described in Nath (1971).
- *BiasCorrectionToolTable,* which generates a table of output.

4.5.3. How to run the BiasCorrectionTool module

The BiasCorrectionTool module can be invoked with the following SAS/IML command:

```
%BiasCorrectionTool( inputData, errorCode, cutOffT1T2 );
```

Table 1 lists the names and descriptions of each of the three input parameters required to run BiasCorrectionTool.

No.	*Argument Name*	*Description*
1.	*inputData*	dataset to be analyzed; see below for specific details on formatting
2.	*cutOffT1T2*	a vector of biopsy threshold values; values should be ordered to match the order of the test scores in the input dataset
3.	*errorCode*	a numeric value for variables that error out; should be set to a number that does not appear in the dataset

Table 1: Description of the parameters used for BiasCorrectionSimulator

The input dataset must contain the following three columns:

- `T1SCORE`, a numeric variable containing participants' screening test scores for one screening test,
- `T2SCORE`, a numeric variable containing participants' screening test scores for the other screening test
- `obsDisease`, a numeric variable with an indicator for participants' observed disease status, 1 for observed cases and 0 for observed non-cases.

For a demonstration of the BiasCorrectionTool module see section 5.

4.5.4. BiasCorrectionTool output

The BiasCorrectionTool module outputs a list of results as a SAS dataset ("resultsBC.sas7bdat"), and a summary report as both a SAS dataset ("BiasCorrectionToolTable.sas7bdat") and a Microsoft Word document ("BiasCorrectionToolTable.doc"). By default, the output is stored in `../Output/<project>/<experiment>`, relative to the "Main" folder. The default location can be changed by altering the file path in the *DefineOutputPath* module.

We provide a variable key for the "resultsBC.sas7bdat" dataset in Appendix A.

The variables in the summary report dataset are as follows:

1. `deltaAUC` is the difference in the AUCs between the two screening tests,
2. `SE` is the standard error of the difference in AUCs,
3. `z` is the z-statistic for the hypothesis test of no difference in AUCs,
4. `p` is the p-value for the hypothesis test of no difference in AUCs, and
5. `analysis` specifies the type of analysis, standard or bias-corrected.

4.6. How to use the DecisionErrorTool

4.6.1. Overview

The DecisionErrorTool module is meant to assist scientists in choosing the most appropriate analysis plan for their paired screening trial. The scientist will input the screening test score means, variances, and correlations they expect to see in their study population. The DecisionErrorTool will simulate data from the study population and output the Type I error rate and power for the standard analysis and the bias-corrected analysis. The scientist can choose the analysis with a nominal Type I error rate and highest power for the correct decision.

In order to calculate the Type I error rate and power, the DecisionErrorTool simulates results for both the null and alternative hypotheses. Results for the alternative hypothesis are calculated using user-entered means, variances, and correlations that assume there will be some difference between the two screening tests. Results for the null hypothesis are calculated assuming that the means, variances, and correlations for Test 2 are equal to the user-provided means, variances, and correlations for Test 1.

The DecisionErrorTool calculates the Type I error rate and power for the user-entered number of realizations of the simulated data. Type I error rate is the proportion of times the null hypothesis is rejected when the null hypothesis is true. Power is the proportion of times the null hypothesis is rejected when the alternative hypothesis is true. The DecisionErrorTool divides power into two fractions: the correct rejection fraction and the wrong rejection fraction. The distinction between fractions is made by comparison to the true area under the curve. The true area under the curve is the area under the curve calculated from the user-entered values for the means, variances, and correlations of the screening test scores. The correct rejection fraction is the proportion of realizations of the data where 1) the alternative hypothesis is true, 2) the hypothesis test rejects, and 3) the screening test with the largest calculated area under the curve is the same as the screening with the largest true area under the curve. The wrong rejection fraction is the proportion of realizations of the data where 1) the alternative hypothesis is true, 2) the hypothesis test rejects, and 3) the screening test with the largest calculated area under the curve is not the screening test with the largest true area under the curve.

4.6.2. DecisionErrorTool algorithm

Algorithm 2 outlines the steps in the DecisionErrorTool.

Algorithm 2 DecisionErrorTool

2.1 Generate a matrix of random data under the alternative hypothesis consisting of the following columns:
  2.1.1 a column of indicators for true disease status (1 = case, 0 = non-case),
  2.1.2 two columns of bivariate Gaussian screening test scores conditional on true disease status,
  2.1.3 a column of signs and symptoms indicators (1 = participant shows signs and symptoms of disease, 0 = otherwise).
2.2 Deduce the observed disease status for each participant as described in **?** (in review).
2.3 Partition participants into true cases and non-cases.
2.4 Calculate the means, variances, and correlations for the true cases and non-cases, called the true means, variances, and correlations.
2.5 Calculate the difference in the areas under the binormal receiver operating characteristic curves (pg. 122, Zhou *et al.* 2002) based on the true means, variances, and correlations of the case and non-case screening test scores.
2.6 Calculate a z-statistic for the reference analysis, which uses the true means, variances, and correlations as inputs. Note: the reference analysis is referred to as the complete analysis in the program since it requires the assumption that the study investigator has complete disease status ascertainment.
2.7 Determine whether the hypothesis test for the reference analysis accepts or rejects.
2.8 Complete Steps 1.1 - 1.12 in Algorithm 1.
2.9 Repeat Steps 1 through 8 for the null hypothesis (means, variances, and correlations for Test 2 equal to those of Test 1).
2.10 Calculate the Type I error, correct rejection fraction, and wrong rejection fraction.
2.11 Generate a report.

To perform the decision error simulations, the DecisionErrorTool contains 8 predefined submodules, as outlined below:

- *completeAnalysis*, which performs the reference analysis, also referred to as the complete analysis (Steps 2.3-2.6).
- *standardAnalysis,* which performs the standard analysis (Steps 1.2-1.5).
- *nathInAllQuadrants,* which calculates the MLEs for the screening test scores of the observed cases in each quadrant. The module first checks that the quadrants contain two or more observations (Steps 1.6-1.7).
- *biasCorrectionAnalysis,* which performs the bias-corrected analysis (*nathInAllQuadrants* and Steps 1.8-1.12).
- *decisionErrorAnalysis,* which generates random data and runs all three analyses (*completeAnalysis*, *standardAnalysis*, *biasCorrectionAnalysis*).
- *setErrorCodes,* which sets an error code for variables based on the error encountered at several steps in the *decisionErrorAnalsis* submodule.
- *DecisionErrorSimulator,* which runs *decisionErrorAnalysis* and outputs a SAS dataset containing the results of the three analyses (reference, standard, and bias-corrected).

In turn, the submodules call a series of stored modules. We give a description of the stored modules in Section 4.5.2. In addition, the DecisionErrorTool calls the following stored module:

- *DecisonErrorToolTable*, which generates a results summary report for the DecisionErrorTool.

4.6.3. How to run the DecisionErrorTool module

The DecisionErrorTool can be invoked via the following SAS/IML command:

```
%DecisionErrorTool( seed ,errC ,nOfreal ,samSize ,mT1ScoC,mT2ScoC,stdT1ScoC ,
                    stdT2ScoC ,corrT1T2C ,mT1ScoNonC,mT2ScoNonC,stdT1ScoNonC ,
                    stdT2ScoNonC ,corrT1T2NonC ,disPrev ,cutOffT1T2 ,rOfSS );
```

Table 2 describes the arguments required to run the DecisionErrorTool. The label (Screening Test 1 or Screening Test 2) applied to a particular screening test in the study is arbitrary. However, once the label is applied, take care that the means, variances, correlations, and biopsy thresholds are mapped to the correct label.

No.	***Argument Name***	***Description***
1.	*seed*	seed for generating random data
2.	*errC*	code for undefined entries; set to a value that would not likely appear in the simulated dataset
3.	*nOfreal*	number of realizations of the randomly generated data; typically set to 10000
4.	*samSize*	sample size of planned study
5.	*mT1ScoC*	mean of Screening Test 1 scores for the cases
6.	*mT2ScoC*	mean of Screening Test 2 scores for the cases
7.	*stdT1ScoC*	standard deviation of Screening Test 1 scores for the cases
8.	*stdT2ScoC*	standard deviation of Screening Test 2 scores for the cases
9.	*corrT1T2C*	correlation between Screening Test 1 and 2 scores for the cases
10.	*mT1ScoNonC*	mean of Screening Test 1 scores for the non-cases
11.	*mT2ScoNonC*	mean of Screening Test 2 scores for the non-cases
12.	*stdT1ScoNonC*	standard deviation of Screening Test 1 scores for the non-cases
13.	*stdT2ScoNonC*	standard deviation of Screening Test 2 scores for the non-cases
14.	*corrT1T2NonC*	correlation between Screening Test 1 and 2 scores for the non-cases
15.	*disPrev*	prevalence of disease in the population
16.	*cutOffT1T2*	vector of biopsy thresholds; first list the threshold for Test 1, then Test 2
17.	*rOfSS*	proportion of cases that experience signs and symptoms of disease during the follow-up period

Table 2: List of input parameters to the DecisionErrorTool module.

For a demonstration of the DecisionErrorTool module, see section 5.

4.6.4. DecisionErrorTool output

The DecisionErrorTool outputs a list of results as a SAS dataset ("resultsDE.sas7bdat"), and a summary report as both a SAS dataset ("DecisionErrorToolTable.sas7bdat") and a Microsoft Word document ("DecisionErrorToolTable.doc"). By default, the output is stored in `../Output/<project>/<experiment>`, relative to the "Main" folder. The default location can be changed by altering the file path in the *DefineOutputPath* module.

We provide a variable key for the "resultsDE.sas7bdat" dataset in Appendix B.

The variables in the summary report dataset are as follows:

1. `decisionError` lists the decision type: Type I error, correct rejection, or wrong rejection,

2. `propTrue` is the proportion of realizations of the data where the reference analysis rejects the null hypothesis,
3. `propObs` is the proportion of realizations of the data where the standard analysis rejects the null hypothesis,
4. `propCorr` is the proportion of realizations of the data where the bias-corrected analysis rejects the null hypothesis, and
5. `order` lists the desired row order for the summary report table.

5. A driving example

We use the following prompt to demonstrate the utility of the software. Consider a planned paired oral cancer screening trial comparing the performance of 1) visual, tactile oral exam with biopsy referral for frank cancers (Test 1) and 2) a visual, tactile oral exam, a viewing with the VELscope device, and biopsy referral for lesions observed with either method (Test 2). The design is similar to one considered by Lingen (2009). Participants will be screened with both Test 1 and Test 2. The clinician records their level of suspicion on a scale of 0-100. Participants are considered screen positive if their score is above 65 for Test 1 or 59 for Test 2. Participants who show signs and symptoms of disease during that year are referred for further workup leading to biopsy. Participants who do no show signs and symptoms during the follow-up period are assumed to be disease free. The study investigator plans to enroll 50,000 participants.

The study investigator will use the Bias Correction Suite to 1) determine if correcting for paired screening trial bias would result in a nominal Type I error rate and improved power for the correct decision, and 2) obtain bias-corrected results if bias correction is indicated.

The study investigator will use the DecisionErrorTool to answer the first question. Based on pilot data and previous studies, the study investigator speculates a list of inputs for the DecisionErrorTool. The inputs are listed in Table 3. The parameter values result in an AUC of 0.77 for Test 1 and 0.71 for Test 2.

No.	*Parameter*	*Speculated Value*
1.	Mean of Screening Test 1 Scores for Cases	61.1
2.	Mean of Screening Test 2 Scores for Cases	62.5
3.	Standard Deviation of Screening Test 1 Scores for Cases	1
4.	Standard Deviation of Screening Test 2 Scores for Cases	5
5.	Correlation Between Screening Test 1 and 2 Scores for Cases	0.1
6.	Mean of Screening Test 1 Scores for Non-Cases	60
7.	Mean of Screening Test 2 Scores for Non-Cases	58
8.	Standard Deviation of Screening Test 1 Scores for Non-Cases	1
9.	Standard Deviation of Screening Test 2 Scores for Non-Cases	5
10.	Correlation Between Screening Test 1 and 2 Scores for Non-Cases	0.1
11.	Disease Prevalence	0.01
12.	Rate of Signs and Symptoms	0.1

Table 3: List of speculated values for a hypothetical oral cancer screening trial.

The study investigator prepares the SAS program, *ExampleProgramToRunDecisionErrorTool* to invoke the DecisionErrorTool. We include the program in the Bias Correction Suite files. The example program uses the following set of programming statements:

```
%include "..\Modules\IncludeLibrary.sas";

%IncludeLibrary;

%CommonMain( Project, Experiment, ExampleProgramToRunDecisionErrorTool );

%DecisionErrorTool( 1066, -99999, 10000, 50000, 61.1, 62.5, 1, 5, .1, 60, 58,
                    1, 5, .1, .01, { 65 59 }, .1 );
```

When the study investigator runs *ExampleProgramToRunDecisionErrorTool*, the module creates a dataset, performs analyses, and tabulates results. Figure 2 gives the results of the analysis. The simulated Type I error rate of the bias-corrected analysis is below nominal at 0.03, while the Type I error rate of the standard analysis is above nominal at 0.06. The correct rejection fraction of the bias-corrected analysis is 0.58, while that of the standard analysis is zero. In fact, using the standard analysis, the study investigator would wrongly conclude that Test 2 is superior to Test 1 86% of the time.

Decision Metric	True	Standard	Corrected
Type I Error[1]	0.01	0.06	0.03
Correct Rejection Fraction	0.77	0	0.58
Wrong Rejection Fraction	0	0.86	0.01

[1]Screening Test 1 parameter values are used for both screening tests.

Figure 2: DecisionErrorTool summary report for a planned oral cancer screening study.

Based on the simulation, the study investigator plans to use the bias correction method to correct the results of the hypothetical oral cancer screening study. After collecting data from trial, the study investigator creates an input dataset titled "ExampleDataset". Figure 3 shows the first 20 observations in the dataset and illustrates proper formatting. The variables "T1SCORE" and "T2SCORE" contain the participants' screening test scores for Test 1 and Test 2, respectively. The variable "obsDisease" contains the observed disease status of the study participants.

Obs	T1SCORE	T2SCORE	obsDisease
1	60.64	60.20	1
2	61.40	62.32	1
3	61.33	67.60	1
4	62.06	66.24	1
5	61.89	66.47	1
6	60.53	57.27	0
7	62.60	58.06	0
8	63.15	56.25	0
9	61.98	64.06	1
10	62.82	66.11	1
11	60.83	65.21	1
12	61.43	52.64	0
13	59.29	64.24	1
14	61.51	62.14	1
15	59.89	64.72	1
16	61.15	65.95	1
17	59.72	62.80	1
18	61.62	73.99	1
19	61.09	64.34	1
20	61.84	68.15	1

Figure 3: Example input dataset for the BiasCorrectionTool module.

The study investigator prepares the SAS program, *ExampleProgramToRunBiasCorrectionTool* to invoke the BiasCorrectionTool. We include the example program and input dataset in the Bias Correction Suite files. The example program uses the following set of programming statements:

```
%include "..\Modules\IncludeLibrary.sas";

%IncludeLibrary;

%CommonMain( Project, Experiment, ExampleProgramToRunBiasCorrectionTool );

%BiasCorrectionTool( Example1, -99999, { 65 59 } );
```

When the study investigator runs *ExampleProgramToRunBiasCorrectionTool*, the module inputs the dataset, performs analyses, and tabulates results. Figure 4 shows the results of the standard and bias-corrected analyses. Based on the results of the bias-corrected analysis, the study investigator concludes that Test 1 is the superior screening test.

Analysis	Difference in AUC (Test1 - Test2)	Standard Error	Z	P-Value[1]
Standard	-0.06	0.02	2.79	0.0052
Corrected	0.06	0.02	3.30	0.0010

Figure 4: BiasCorrectionTool summary report for a hypothetical oral cancer screening study.

6. Error handling

If the BiasCorrectionTool or DecisionErrorTool modules reach an unresolvable error (usually by dividing by a number near zero or multiplying by too large a number), they will increment the "cumError" variable but continue to run. If the error causes the module to be unable to compute one of the output variables, the output matrix will contain a number of columns less than the required amount (107 for the BiasCorrectionTool and 164 for the DecisionErrorTool). In this case, the module appends additional columns at the end of the matrix to reach the required number of columns. The IML output matrix must contain the required number of columns in order for the output to be appended to the SAS output dataset. If the variable "cumError" is not equal to zero, then the results of the analysis may be invalid and bias correction may not be possible for the input dataset or input parameters.

7. Additional resources

Additional information about the Bias Correction Suite can be found at our website, www.samplesizeshop.org. The Sample Size Shop project is a collaborative effort between the University of Florida and the University of Colorado Denver. The goals of the project are to develop new statistical methods for calculating power and sample size for the general linear multivariate model and the linear mixed model, provide user-friendly software to perform power and sample size calculations, and educate researchers regarding both the methods and the software.

References

Glueck DH, Lamb MM, O'Donnell CI, Ringham BM, Brinton JT, Muller KE, Lewin JM, Alonzo TA, Pisano ED (2009). "Bias in trials comparing paired continuous tests can cause researchers to choose the wrong screening modality." *BMC Medical Research Methodology*, **9**, 4. ISSN 1471-2288. doi:10.1186/1471-2288-9-4. PMID: 19154609, URL http://www.ncbi.nlm.nih.gov/pubmed/19154609.

Lingen MW (2009). *Efficacy of oral cancer screening adjunctive techniques*. National Institute of Dental and Craniofacial Research, National Institutes of Health, US Department of Health and Human Services; (NIH Project Number: 1RC2DE020779-01), Bethesda, MD.

Nath GB (1971). "Estimation in Truncated Bivariate Normal Distributions." *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, **20**(3), 313–319. ISSN 0035-9254. doi:10.2307/2346762. ArticleType: research-article / Full publication date: 1971 / Copyright Â© 1971 Royal Statistical Society, URL http://www.jstor.org/stable/2346762.

Obuchowski NA, McClish DK (1997). "Sample size determination for diagnostic accuracy studies involving binormal ROC curve indices." *Statistics in Medicine*, **16**(13), 1529–1542. ISSN 0277-6715. PMID: 9249923, URL http://www.ncbi.nlm.nih.gov/pubmed/9249923.

Zhou XH, McClish DK, Obuchowski NA (2002). *Statistical Methods in Diagnostic Medicine*. 1 edition. Wiley-Interscience. ISBN 0471347728.

A. BiasCorrectionTool Variable Key

Table A.1 lists the name and description for each output variable in the "resultsBC.sasbdat7" dataset.

Table A.1: BiasCorrectionTool output variables.

Parameter name	Description
obsError	Less than 2 observed cases or non-cases
nathError	Nath algorithm did not converge in any quadrant
weighted	Source of corrected estimates (1 = weighted, 0 = nath)
correctedNumberCasesError	Corrected Number of Cases Exceeded Total Sample Size
empty123	Less than 2 observed cases in quadrants 1, 2, and 3 together
emptyQuad1	Less than 2 observed cases in quadrant 1
emptyQuad2	Less than 2 observed cases in quadrant 2
emptyQuad3	Less than 2 observed cases in quadrant 3
emptyQuad4	Less than 2 observed cases in quadrant 4
cutOffTest1	Cutoff for Test 1
cutOffTest2	Cutoff for Test 2
nathMaxIterations	Maximum Number of Iterations for Nath Algorithm
nathTolerance	Tolerance for Nath Algorithm
sampleSize	Sample Size
numberObsCases	Number of Observed Cases
meanT1ObsCases	Test 1 Mean for Observed Cases
meanT2ObsCases	Test 2 Mean for Observed Cases
stdT1ObsCases	Test 1 Standard Deviation for Observed Cases
stdT2ObsCases	Test 2 Standard Deviation for Observed Cases
corrObsCases	Correlation Between Observed Cases
numberObsNonCases	Number of Observed Non-Cases
meanT1ObsNonCases	Test 1 Mean for Observed Non-Cases
meanT2ObsNonCases	Test 2 Mean for Observed Non-Cases
stdT1ObsNonCases	Test 1 Standard Deviation for Observed Non-Cases
stdT2ObsNonCases	Test 2 Standard Deviation for Observed Non-Cases
corrObsNonCases	Correlation Between Observed Non-Cases
numberObsCasesQuadrant1	Number of Observed Cases in Quadrant 1
meanT1ObsCasesQuadrant1	Test 1 Mean for Observed Cases in Quadrant 1
meanT2ObsCasesQuadrant1	Test 2 Mean for Observed Cases in Quadrant 1
stdT1ObsCasesQuadrant1	Test 1 Standard Deviation for Observed Cases in Quadrant 1
stdT2ObsCasesQuadrant1	Test 2 Standard Deviation for Observed Cases in Quadrant 1
corrObsCasesQuadrant1	Correlation Between Observed Cases in Quadrant 1
numberObsCasesQuadrant2	Number of Observed Cases in Quadrant 2
meanT1ObsCasesQuadrant2	Test 1 Mean for Observed Cases in Quadrant 2
meanT2ObsCasesQuadrant2	Test 2 Mean for Observed Cases in Quadrant 2
stdT1ObsCasesQuadrant2	Test 1 Standard Deviation for Observed Cases in Quadrant 2
stdT2ObsCasesQuadrant2	Test 2 Standard Deviation for Observed Cases in Quadrant 2
corrObsCasesQuadrant2	Correlation Between Observed Cases in Quadrant 2
numberObsCasesQuadrant3	Number of Observed Cases in Quadrant 3
meanT1ObsCasesQuadrant3	Test 1 Mean for Observed Cases in Quadrant 3

meanT2ObsCasesQuadrant3	Test 2 Mean for Observed Cases in Quadrant 3
stdT1ObsCasesQuadrant3	Test 1 Standard Deviation for Observed Cases in Quadrant 3
stdT2ObsCasesQuadrant3	Test 2 Standard Deviation for Observed Cases in Quadrant 3
corrObsCasesQuadrant3	Correlation Between Observed Cases in Quadrant 3
numberObsCasesQuadrant4	Number of Observed Cases in Quadrant 4
meanT1ObsCasesQuadrant4	Test 1 Mean for Observed Cases in Quadrant 4
meanT2ObsCasesQuadrant4	Test 2 Mean for Observed Cases in Quadrant 4
stdT1ObsCasesQuadrant4	Test 1 Standard Deviation for Observed Cases in Quadrant 4
stdT2ObsCasesQuadrant4	Test 2 Standard Deviation for Observed Cases in Quadrant 4
corrObsCasesQuadrant4	Correlation Between Observed Cases in Quadrant 4
numberObsCasesQuadrant4	Number of Observed Cases in Quadrant 4
meanT1ObsCasesQuadrant4	Test 1 Mean for Observed Cases in Quadrant 4
meanT2ObsCasesQuadrant4	Test 2 Mean for Observed Cases in Quadrant 4
stdT1ObsCasesQuadrant4	Test 1 Standard Deviation for Observed Cases in Quadrant 4
stdT2ObsCasesQuadrant4	Test 2 Standard Deviation for Observed Cases in Quadrant 4
corrObsCasesQuadrant4	Correlation Between Observed Cases in Quadrant 4
numberObsCasesQ123	Number of Observed Cases in Quadrants 1, 2, and 3 Together
meanT1ObsCasesQ123	Test 1 Mean for Observed Cases in Quadrants 1, 2, and 3 Together
meanT2ObsCasesQ123	Test 2 Mean for Observed Cases in Quadrants 1, 2, and 3 Together
stdT1ObsCasesQ123	Test 1 Standard Deviation for Observed Cases in Quadrants 1, 2, and 3 Together
stdT2ObsCasesQ123	Test 2 Standard Deviation for Observed Cases in Quadrants 1, 2, and 3 Together
corrObsCasesQ123	Correlation Between Observed Cases in Quadrants 1, 2, and 3 Together
outcomeNathQuad1	Outcome of Nath Algorithm in Quadrant 1
outcomeNathQuad2	Outcome of Nath Algorithm in Quadrant 2
outcomeNathQuad3	Outcome of Nath Algorithm in Quadrant 3
outcomeNathQuad4	Outcome of Nath Algorithm in Quadrant 4
choice	Nath Estimates Chosen From This Quadrant
logL	Log Likelihood for Nath Estimates from Chosen Quadrant
nathIterations	Number of Iterations of Nath Algorithm for Chosen Quadrant
nathOutcome	Nath Error Code for Chosen Quadrant
nathMu1	Nath Estimate for Mu1
nathMu2	Nath Estimate for Mu2
nathSig1	Nath Estimate for Sig1
nathSig2	Nath Estimate for Sig2
nathRho	Nath Estimate for Rho
probQ123Nath	Estimated Probability of Quadrants 1, 2, and 3 Based On Nath
probQuadrant4Nath	Estimated Probability of Quadrant 4 Based On Nath
probQ123Weighted	Estimated Probability of Quadrants 1, 2, and 3 Based On Weighted Estimates
probQuadrant4Weighted	Estimated Probability of Quadrant 4 Based On Weighted Estimates
correctedNumberCases	Corrected Number of Cases Based On Nath
correctedNumberNonCases	Corrected Number of Non-Cases Based On Nath
weightedMu1	Weighted Estimate for Mu1

weightedMu2	Weighted Estimate for Mu2
weightedSig1	Weighted Estimate for sig1
weightedSig2	Weighted Estimate for sig2
weightedRho	Weighted Estimate for Rho
correctedMu1	Corrected Estimate for Mu1
correctedMu2	Corrected Estimate for Mu2
correctedSig1	Corrected Estimate for Sig1
correctedSig2	Corrected Estimate for Sig2
correctedRho	Corrected Estimate for Rho
observedAUCT1ParmA	Parameter A for Test 1 Binormal AUC Calculated Using Observed Summary Stats
observedAUCT1ParmB	Parameter B for Test 1 Binormal AUC Calculated Using Observed Summary Stats
observedAUCT1	Test 1 AUC Calculated From Observed Summary Stats
observedAUCT2ParmA	Parameter A for Test 2 Binormal AUC Calculated Using Observed Summary Stats
observedAUCT2ParmB	Parameter B for Test 2 Binormal AUC Calculated Using Observed Summary Stats
observedAUCT2	Test 2 AUC Calculated From Observed Summary Stats
observedDeltaAUC	Difference in AUC (Test 1 - Test 2) Calculated From Observed Summary Stats
observedSEDeltaAUC	Standard Error of the Difference in AUC Calculated From Observed Summary Stats
observedZDeltaAUC	Z Value for Delta AUC Hypothesis Test Calculated From Observed Summary Stats
observedPDeltaAUC	P Value for Delta AUC Hypothesis Test Calculated From Observed Summary Stats
observedReject	Indicator that Delta AUC Hypothesis Test Calculated From Observed Summary Stats Rejected (1 = reject, 0 = fail to reject)
correctedAUCT1ParmA	Parameter A for Test 1 Binormal AUC Calculated Using Corrected Summary Stats for Cases / Observed for Non-Cases
correctedAUCT1ParmB	Parameter B for Test 1 Binormal AUC Calculated Using Corrected Summary Stats for Cases / Observed for Non-Cases
correctedAUCT1	Test 1 AUC Calculated From Corrected Summary Stats for Cases / Observed for Non-Cases
correctedAUCT2ParmA	Parameter A for Test 2 Binormal AUC Calculated Using Corrected Summary Stats for Cases / Observed for Non-Cases
correctedAUCT2ParmB	Parameter B for Test 2 Binormal AUC Calculated Using Corrected Summary Stats for Cases / Observed for Non-Cases
correctedAUCT2	Test 2 AUC Calculated From Corrected Summary Stats for Cases / Observed for Non-Cases
correctedDeltaAUC	Difference in AUC (Test 1 - Test 2) Calculated From Corrected Summary Stats for Cases / Observed for Non-Cases
correctedSEDeltaAUC	Standard Error of the Difference in AUC Calculated From Corrected Summary Stats for Cases / Observed for Non-Cases

correctedZDeltaAUC	Z Value for Delta AUC Hypothesis Test Calculated From Corrected Summary Stats for Cases / Observed for Non-Cases
correctedPDeltaAUC	P Value for Delta AUC Hypothesis Test Calculated From Corrected Summary Stats for Cases / Observed for Non-Cases
correctedReject	Indicator that Delta AUC Hypothesis Test Calculated From Corrected Summary Stats for Cases / Observed for Non-Cases Rejected (1 = reject, 0 = fail to reject)
additionalColumnsNeeded	Number of empty columns added to the row to make the matrix conform - if this number is anything but 0 DO NOT USE ROW
columnError	1 = program errored out and dummy columns were created, 0 = no errors that required dummy columns
resumeError	1 = program errored out and resumed without resolving error, 0 = no resume error
cumErrors	Cumulative number of errors ignored during program

B. DecisionErrorTool Variable Key

Table B.1 lists the name and description for each output variable in the "resultsDE.sasbdat7" dataset.

Table B.1: DecisionErrorTool output variables.

Parameter name	Description
rep	Dataset ID
vectorRow	Experimental Variable Index
trueError	Less than 2 cases or non-cases
obsError	Less than 2 observed cases or non-cases
nathError	Nath algorithm did not converge in any quadrant
weighted	Source of corrected estimates (1 = weighted, 0 = nath)
correctedNumberCasesError	Corrected Number of Cases Exceeded Total Sample Size
hypErrorComplete	Population Delta AUC = 0 But Observed Hypothesis Test Rejects
hypErrorObserved	Population Delta AUC = 0 But Observed Hypothesis Test Rejects
hypErrorCorrected	Population Delta AUC = 0 But Corrected Hypothesis Test Rejects
empty123	Less than 2 observed cases in quadrants 1, 2, and 3 together
emptyQuad1	Less than 2 observed cases in quadrant 1
emptyQuad2	Less than 2 observed cases in quadrant 2
emptyQuad3	Less than 2 observed cases in quadrant 3
emptyQuad4	Less than 2 observed cases in quadrant 4
diseasePrevalance	Disease Prevalence
rateSignsSymptoms	Rate of Signs and Symptoms
cutOffTest1	Cutoff for Test 1
cutOffTest2	Cutoff for Test 2
populationRhoCases	True Value of Rho for Cases
populationRhoNonCases	True Value of Rho for Non-Cases
nathMaxIterations	Maximum Number of Iterations for Nath Algorithm
nathTolerance	Tolerance for Nath Algorithm
sampleSize	Sample Size
populationMu1Cases	True Value of Mu1 for Cases
populationMu2Cases	True Value of Mu2 for Cases
populationMu1NonCases	True Value of Mu1 for Non-Cases
populationMu2NonCases	True Value of Mu2 for Non-Cases
populationSig1Cases	True Value of Sigma1 for Cases
populationSig2Cases	True Value of Sigma2 for Cases
populationSig1NonCases	True Value of Sigma1 for Non-Cases
populationSig2NonCases	True Value of Sigma2 for Non-Cases
quadrantMLECases	Quadrant Containing the MLE for Cases
quadrantMLENonCases	Quadrant Containing the MLE for Non-Cases
numberCases	Number of True Cases
meanT1Cases	Test 1 Mean for True Cases
meanT2Cases	Test 2 Mean for True Cases
stdT1Cases	Test 1 Standard Deviation for True Cases
stdT2Cases	Test 2 Standard Deviation for True Cases
corrCases	Correlation Between True Cases

numberNonCases	Number of True Non-Cases
meanT1NonCases	Test 1 Mean for True Non-Cases
meanT2NonCases	Test 2 Mean for True Non-Cases
stdT1NonCases	Test 1 Standard Deviation for True Non-Cases
stdT2NonCases	Test 2 Standard Deviation for True Non-Cases
corrNonCases	Correlation Between True Non-Cases
numberObsCases	Number of Observed Cases
meanT1ObsCases	Test 1 Mean for Observed Cases
meanT2ObsCases	Test 2 Mean for Observed Cases
stdT1ObsCases	Test 1 Standard Deviation for Observed Cases
stdT2ObsCases	Test 2 Standard Deviation for Observed Cases
corrObsCases	Correlation Between Observed Cases
numberObsNonCases	Number of Observed Non-Cases
meanT1ObsNonCases	Test 1 Mean for Observed Non-Cases
meanT2ObsNonCases	Test 2 Mean for Observed Non-Cases
stdT1ObsNonCases	Test 1 Standard Deviation for Observed Non-Cases
stdT2ObsNonCases	Test 2 Standard Deviation for Observed Non-Cases
corrObsNonCases	Correlation Between Observed Non-Cases
posT1SSGoldStd	Proportion of Cases That Screen Positive on Test 1 or Screen Negative on Both Tests and Show Signs and Symptoms
posT2SSGoldStd	Proportion of Cases That Screen Positive on Test 2 or Screen Negative on Both Tests and Show Signs and Symptoms
GoldStd	Proportion of Cases That Receive the Gold Standard For Any Reason
numberObsCasesQuadrant1	Number of Observed Cases in Quadrant 1
meanT1ObsCasesQuadrant1	Test 1 Mean for Observed Cases in Quadrant 1
meanT2ObsCasesQuadrant1	Test 2 Mean for Observed Cases in Quadrant 1
stdT1ObsCasesQuadrant1	Test 1 Standard Deviation for Observed Cases in Quadrant 1
stdT2ObsCasesQuadrant1	Test 2 Standard Deviation for Observed Cases in Quadrant 1
corrObsCasesQuadrant1	Correlation Between Observed Cases in Quadrant 1
numberObsCasesQuadrant2	Number of Observed Cases in Quadrant 2
meanT1ObsCasesQuadrant2	Test 1 Mean for Observed Cases in Quadrant 2
meanT2ObsCasesQuadrant2	Test 2 Mean for Observed Cases in Quadrant 2
stdT1ObsCasesQuadrant2	Test 1 Standard Deviation for Observed Cases in Quadrant 2
stdT2ObsCasesQuadrant2	Test 2 Standard Deviation for Observed Cases in Quadrant 2
corrObsCasesQuadrant2	Correlation Between Observed Cases in Quadrant 2
numberObsCasesQuadrant3	Number of Observed Cases in Quadrant 3
meanT1ObsCasesQuadrant3	Test 1 Mean for Observed Cases in Quadrant 3
meanT2ObsCasesQuadrant3	Test 2 Mean for Observed Cases in Quadrant 3
stdT1ObsCasesQuadrant3	Test 1 Standard Deviation for Observed Cases in Quadrant 3
stdT2ObsCasesQuadrant3	Test 2 Standard Deviation for Observed Cases in Quadrant 3
corrObsCasesQuadrant3	Correlation Between Observed Cases in Quadrant 3
numberObsCasesQuadrant4	Number of Observed Cases in Quadrant 4
meanT1ObsCasesQuadrant4	Test 1 Mean for Observed Cases in Quadrant 4
meanT2ObsCasesQuadrant4	Test 2 Mean for Observed Cases in Quadrant 4
stdT1ObsCasesQuadrant4	Test 1 Standard Deviation for Observed Cases in Quadrant 4

stdT2ObsCasesQuadrant4	Test 2 Standard Deviation for Observed Cases in Quadrant 4
corrObsCasesQuadrant4	Correlation Between Observed Cases in Quadrant 4
numberObsCasesQ123	Number of Observed Cases in Quadrants 1, 2, and 3 Together
meanT1ObsCasesQ123	Test 1 Mean for Observed Cases in Quadrants 1, 2, and 3 Together
meanT2ObsCasesQ123	Test 2 Mean for Observed Cases in Quadrants 1, 2, and 3 Together
stdT1ObsCasesQ123	Test 1 Standard Deviation for Observed Cases in Quadrants 1, 2, and 3 Together
stdT2ObsCasesQ123	Test 2 Standard Deviation for Observed Cases in Quadrants 1, 2, and 3 Together
corrObsCasesQ123	Correlation Between Observed Cases in Quadrants 1, 2, and 3 Together
outcomeNathQuad1	Outcome of Nath Algorithm in Quadrant 1
outcomeNathQuad2	Outcome of Nath Algorithm in Quadrant 2
outcomeNathQuad3	Outcome of Nath Algorithm in Quadrant 3
outcomeNathQuad4	Outcome of Nath Algorithm in Quadrant 4
choice	Nath Estimates Chosen From This Quadrant
logL	Log Likelihood for Nath Estimates from Chosen Quadrant
nathIterations	Number of Iterations of Nath Algorithm for Chosen Quadrant
nathOutcome	Nath Error Code for Chosen Quadrant
nathMu1	Nath Estimate for Mu1
nathMu2	Nath Estimate for Mu2
nathSig1	Nath Estimate for Sig1
nathSig2	Nath Estimate for Sig2
nathRho	Nath Estimate for Rho
probQ123Nath	Estimated Probability of Quadrants 1, 2, and 3 Based On Nath
probQuadrant4Nath	Estimated Probability of Quadrant 4 Based On Nath
probQ123Weighted	Estimated Probability of Quadrants 1, 2, and 3 Based On Weighted Estimates
probQuadrant4Weighted	Estimated Probability of Quadrant 4 Based On Weighted Estimates
correctedNumberCases	Corrected Number of Cases Based On Nath
correctedNumberNonCases	Corrected Number of Non-Cases Based On Nath
weightedMu1	Weighted Estimate for Mu1
weightedMu2	Weighted Estimate for Mu2
weightedSig1	Weighted Estimate for Sig1
weightedSig2	Weighted Estimate for Sig2
weightedRho	Weighted Estimate for Rho
correctedMu1	Corrected Estimate for Mu1
correctedMu2	Corrected Estimate for Mu2
correctedSig1	Corrected Estimate for Sig1
correctedSig2	Corrected Estimate for Sig2
correctedRho	Corrected Estimate for Rho
populationAUCT1ParmA	Parameter A for Test 1 Binormal AUC Calculated From Population (True) Values
populationAUCT1ParmB	Parameter B for Test 1 Binormal AUC Calculated From Population (True) Values
populationAUCT1	Test 1 AUC Calculated From Population (True) Values

populationAUCT2ParmA	Parameter A for Test 2 Binormal AUC Calculated From Population (True) Values
populationAUCT2ParmB	Parameter B for Test 2 Binormal AUC Calculated From Population (True) Values
populationAUCT2	Test 2 AUC Calculated From Population (True) Values
populationDeltaAUC	Difference in AUC (Test 1 - Test 2) Calculated From Population (True) Values
completeAUCT1ParmA	Parameter A for Test 1 Binormal AUC Calculated Assuming Complete Disease Status Ascertainment
completeAUCT1ParmB	Parameter B for Test 1 Binormal AUC Calculated Assuming Complete Disease Status Ascertainment
completeAUCT1	Test 1 AUC Calculated Assuming Complete Disease Status Ascertainment
completeAUCT2ParmA	Parameter A for Test 2 Binormal AUC Calculated Assuming Complete Disease Status Ascertainment
completeAUCT2ParmB	Parameter B for Test 2 Binormal AUC Calculated Assuming Complete Disease Status Ascertainment
completeAUCT2	Test 2 AUC Calculated Assuming Complete Disease Status Ascertainment
completeDeltaAUC	Difference in AUC (Test 1 - Test 2) Calculated Assuming Complete Disease Status Ascertainment
completeSEDeltaAUC	Standard Error of the Difference in AUC Assuming Complete Disease Status Ascertainment
completeZDeltaAUC	Z Value for Delta AUC Hypothesis Test Assuming Complete Disease Status Ascertainment
completePDeltaAUC	P Value for Delta AUC Hypothesis Test Assuming Complete Disease Status Ascertainment
completeReject	Indicator that Delta AUC Hypothesis Test Assuming Complete Disease Status Ascertainment Rejected
directionCompleteHypTest	Indicator for Direction of Rejection of Complete Hypothesis Test (0 = fail, 1 = correct, -1 = reverse)
observedAUCT1ParmA	Parameter A for Test 1 Binormal AUC Calculated Using Observed Summary Stats
observedAUCT1ParmB	Parameter B for Test 1 Binormal AUC Calculated Using Observed Summary Stats
observedAUCT1	Test 1 AUC Calculated From Observed Summary Stats
observedAUCT2ParmA	Parameter A for Test 2 Binormal AUC Calculated Using Observed Summary Stats
observedAUCT2ParmB	Parameter B for Test 2 Binormal AUC Calculated Using Observed Summary Stats
observedAUCT2	Test 2 AUC Calculated From Observed Summary Stats
observedDeltaAUC	Difference in AUC (Test 1 - Test 2) Calculated From Observed Summary Stats
observedSEDeltaAUC	Standard Error of the Difference in AUC Calculated From Observed Summary Stats

observedZDeltaAUC	Z Value for Delta AUC Hypothesis Test Calculated From Observed Summary Stats
observedPDeltaAUC	P Value for Delta AUC Hypothesis Test Calculated From Observed Summary Stats
observedReject	Indicator that Delta AUC Hypothesis Test Calculated From Observed Summary Stats Rejected
bias	Bias
directionObservedHypTest	Indicator for Direction of Rejection of Observed Hypothesis Test (0 = fail, 1 = correct, -1 = reverse)
correctedAUCT1ParmA	Parameter A for Test 1 Binormal AUC Calculated Using Corrected Summary Stats for Cases/Observed for Non-Cases
correctedAUCT1ParmB	Parameter B for Test 1 Binormal AUC Calculated Using Corrected Summary Stats for Cases/Observed for Non-Cases
correctedAUCT1	Test 1 AUC Calculated From Corrected Summary Stats for Cases / Observed for Non-Cases
correctedAUCT2ParmA	Parameter A for Test 2 Binormal AUC Calculated Using Corrected Summary Stats for Cases/Observed for Non-Cases
correctedAUCT2ParmB	Parameter B for Test 2 Binormal AUC Calculated Using Corrected Summary Stats for Cases/Observed for Non-Cases
correctedAUCT2	Test 2 AUC Calculated From Corrected Summary Stats for Cases / Observed for Non-Cases
correctedDeltaAUC	Difference in AUC (Test 1 - Test 2)
correctedSEDeltaAUC	Standard Error of the Difference in AUC
correctedZDeltaAUC	Z Value for Delta AUC Hypothesis Test
correctedPDeltaAUC	P Value for Delta AUC Hypothesis Test
correctedReject	Indicator that Delta AUC Hypothesis Test
uncorrectedBias	Uncorrected bias
directionCorrectedHypTest	Indicator for Direction of Rejection of Corrected Hypothesis Test
additionalColumnsNeeded	Number of empty columns added to the row to make the matrix conform
columnError	1 = program errored out, 0 = no errors that required dummy columns
resumeError	1 = program errored out and resumed, 0 = no resume error
errorsThisRep	Number of errors ignored for the current iteration
cumErrors	Cumulative number of errors ignored during program