Paper 440-2013

A SAS® Macro for Computing Pooled Likelihood Ratio Tests with Multiply Imputed Data

Stephen A. Mistler, Arizona State University

ABSTRACT

For multilevel analyses (e.g., linear mixed models), researchers are often interested in pooling, interpreting, and testing both fixed effects and random effects. PROC MIANALYZE has two shortcomings in this regard. First, it cannot easily pool variance estimates. Second, the significance tests of these estimates are Wald-type tests that are inappropriate for testing variance estimates. Likelihood ratio testing is a more flexible approach, as it can be used to compare models that differ in both fixed and random effects. The likelihood ratio test statistic requires a complex calculation that is not included in PROC MIANALYZE. This paper describes a SAS macro, MMI_ANALYZE, that fits two user-specified models in PROC MIXED, pools the estimates from those models (including variance components), and implements a pooled likelihood ratio test.

INTRODUCTION

Researchers commonly use multiple imputation as a method for handling missing data. Multiple imputation can be used to increase power and reduce bias for analyses with missing data if the data are missing at random (MAR). The MI procedure in SAS is used to impute missing data using single-level estimation methods (e.g., linear regression). PROC MI produces multiple data sets, each with a different set of imputed values for the missing data. An analyst can perform an identical analysis on each of the imputed data sets and then pool the results using the MIANALYZE procedure. In most cases, PROC MIANALYZE works well for pooling multiply imputed data.

However, researchers conducting multilevel analyses (e.g., linear mixed models) are often interested in pooling, interpreting, and testing both fixed effects and random effects. The MIANALYZE procedure can be used to easily pool the fixed effects (using the PARMS= option) and the covariance matrix of fixed effects (using the COVB= option). However, PROC MIANALYZE has no built-in functionality for pooling random effects (the G matrix produced by the MIXED procedure). Though PROC MIANALYZE can be used to pool the random effects through a workaround, it is more difficult than pooling the fixed effects. The macro presented in this article, MMI_ANALYZE, can be used to pool both fixed and random effects.

A separate issue is that PROC MIANALYZE performs significance testing through Wald-type tests. These tests are appropriate for testing fixed effects, but are not appropriate for testing random effect estimates (Singer & Willett, 2003). Likelihood ratio testing is the recommended approach to testing random effects, and can be used to compare models that differ in both fixed and random effects. Calculation of the likelihood ratio test statistic requires a complex calculation that is not included in PROC MIANALYZE (Meng and Rubin, 1992). The macro implements this calculation to produce valid likelihood ratio tests for nested models.

OVERVIEW OF MULTIPLE IMPUTATION

A multiple imputation (MI) analysis consists of two distinct phases: the imputation phase and the analysis/pooling phase. In the imputation phase, a Markov Chain Monte Carlo (MCMC) algorithm simulates the distribution of the parameters in the imputation model and uses these parameter estimates to fill in the missing data. This process amounts to sampling missing values from a posterior predictive distribution. The imputation process generates *m* complete data sets, where *m* is a number chosen by the analyst. These imputed data sets are then analyzed in the analysis/pooling phase. This document focuses on the analysis/pooling phase of multilevel multiple imputation. For more information regarding the imputation phase of multilevel multiple imputation, see Mistler (2013).

ANALYSIS AND POOLING

The analysis phase of multiple imputation involves performing the same analysis once for each of the *m* datasets generated by the imputation procedure. If using PROC MIXED in SAS, the addition of a BY statement with the name of the variable containing the imputation number (_IMPUTATION_) tells SAS to perform the same analysis for every imputation. SAS then outputs separate results for each imputation.

The pooling of estimates for multilevel data is identical to the pooling of results for single-level data, and as such is discussed only briefly here. A more thorough discussion of the topic is available in Rubin (1987). Because the analysis yields *m* estimates of each parameter, it is necessary to pool them into a single point estimate. This is done by averaging the parameter estimates across the *m* datasets

$$\bar{\theta} = \frac{1}{m}\sum_{t=1}^{m}\hat{\theta}_t \tag{1}$$

where $\hat{\theta}_t$ is a parameter estimate (e.g., regression coefficient, variance estimate) from data set *t*. Standard errors are slightly more difficult to combine, as the pooled standard error is the combination of the sampling error that would be present if the data were complete (within-imputation variance) and the sampling error resulting from missing data (between-imputation variance). The within-imputation variance is merely the average of the sampling variances, as follows:

$$V_W = \frac{1}{m}\sum_{t=1}^{m}SE_t^2 \tag{2}$$

where SE_t^2 is the squared standard error from data set *t*. The between-imputation variance is calculated as the variance of a parameter across the imputations.

$$V_B = \frac{1}{m-1}\sum_{t=1}^{m}\left(\hat{\theta}_t - \bar{\theta}\right)^2 \tag{3}$$

The total variance is calculated as the combination of the within-imputation and between-imputation variances.

$$V_T = V_W + V_B + \frac{V_B}{m} \tag{4}$$

Finally, the standard error of the parameter is the square root of the total variance.

$$SE = \sqrt{V_T} \tag{5}$$

It is possible to perform a multilevel analysis using PROC MIXED and then use a single-level pooling procedure (e.g., PROC MIANALYZE) to pool the results. However, the process can be difficult and time consuming because SAS only pools certain effects (e.g., regression coefficients). To address this problem, I created a macro, MMI_ANALYZE, which performs analyses in PROC MIXED and then pools the results.

EXAMPLE DATA

To demonstrate the use of the MMI_ANALYZE macro, I use an artificial data set that is loosely based on a math problem solving intervention described by Montague, Enders, and Dietz (2011). The data set mimics a scenario where researchers randomly assign 40 schools, each with 25 students, to an intervention and a control condition (i.e., a cluster-randomized design). In addition to a level-2 intervention code (TX; 0 = control, 1 = intervention), the data set contains a number of student-level covariates, including standardized math scores (MATH), math self-efficacy scores (SELFE), problem-solving pre-test scores (PREPSOLVE), a binary lunch assistance indicator (FRLUNCH), and a binary ethnicity indicator (MINORITY). The outcome variable is end-of-year math problem-solving test scores (PSTPSOLVE). The end-of-year math problem-solving test scores and the self-efficacy ratings are each missing completely at random for 20% of students in the dataset. I use variables from this data set to illustrate the use of the macro. For readers that are interested in duplicating the illustrative analyses, the data set is available upon request.

The data have already been imputed using a macro called MMI_IMPUTE (Mistler, 2013). The MMI_IMPUTE macro performs multilevel multiple imputation, which accounts for clustering in the imputation model. All of the complete variables (TX, MATH, FRLUNCH, MINRITY, and PREPSOLVE) were used to impute the incomplete variables (SELFE and PSTPSOLVE). The imputation model included the intercept as a random effect, but none of the variables were included as random effects. For more information about the macro and for the syntax used to impute the data, see Mistler (2013).

LIKELIHOOD RATIO TESTING AND THE D$_3$ STATISTIC

An essential part of analyzing multilevel data is the comparison of nested models using multiparameter significance tests (i.e., likelihood ratio tests). A restricted model is said to be nested inside of a full model if the restricted model can be created by removing parameters from the full model (or, more accurately, by constraining parameters in the full model to zero). As an example, let's say that we wish to predict PSTPSOLVE from MATH, SELFE, and TX in a multilevel model. The multilevel regression equation for this model is:

$$\text{PSTPSOLVE}_{ij} = \beta_0 + \beta_1(\text{MATH}_{ij}) + \beta_2(\text{SELFE}_{ij}) + \beta_3(\text{TX}_j) + b_{0j} + \varepsilon_{ij} \quad (6)$$

where PSTPSOLVE$_{ij}$ is the outcome for case *i* in cluster *j*. β_0 is the intercept. β_1, β_2, and β_3 are slope coefficients. b_{0j} is a level-2 residual that allows the intercepts to vary across clusters, and ε_{ij} is the within-cluster residual for case *i*. This equation constitutes our full model. Any model predicting PSTPSOLVE from a subset of the aforementioned predictors would be nested within the previous model. For example, a model predicting PSTPSOLVE from only the fixed and random effects of the intercept would be nested within our full analysis model. We can show this by substituting a zero for each slope coefficient (β_1, β_2, and β_3) in Equation 6. After removing the terms that now equal zero, we are left with:

$$\text{PSTPSOLVE}_{ij} = \beta_0 + b_{0j} + \varepsilon_{ij} \quad (7)$$

In the context of a complete-data analysis, likelihood ratio tests can be used to test the significance of multiple variables by computing twice the difference in the log likelihood values from two nested models, which has an approximate chi-square distribution. These tests are often used to test the significance of groups of variables, as in a hierarchical entry procedure from multiple regression. Perhaps most importantly, in multilevel analysis likelihood ratio tests are the preferred method for testing random effects (Singer & Willett, 2003).

The MIANALYZE procedure allows users to perform multiparameter significance testing using Wald tests, but does not provide likelihood ratio tests. The likelihood ratio test cannot simply be averaged like most parameters. Correct pooling of the likelihood ratio test requires a tedious series of calculations to create a new test statistic known as D_3 (developed by Meng and Rubin, 1992; labeled as D_3 by Schafer, 1997). The primary difficulty in calculating the D_3 statistic lies in calculating the log-likelihood for each model with its parameters constrained to the pooled estimates. PROC MIXED cannot be used to calculate these statistics, so they have to be calculated manually. The following is a general outline of the procedure for calculating the D_3 test.

1. Estimate the full and restricted models for each of the *m* data sets, then pool the resulting estimates: $\bar{\theta}_F$ and $\bar{\theta}_R$, respectively.

2. Calculate the likelihood ratio test for each of the *m* datasets:

$$LR = -2(logL_{Restricted} - logL_{Full}) \quad (8)$$

3. Average the likelihood ratio tests across the *m* imputed datasets:

$$\overline{LR} = \frac{1}{m}\sum_{l=1}^{m} LR_l \quad (9)$$

4. Re-estimate the full and restricted models, with their model parameters constrained to the values in $\bar{\theta}_F$ and $\bar{\theta}_R$, respectively. As an alternative to estimating the constrained models, it is possible to directly calculate the likelihood values for each of the two constrained models. Regardless of the method used to calculate the constrained likelihoods, the likelihood ratio test in Equation 10 is subsequently calculated for the two constrained models for each of the *m* datasets.

$$LR_{Constrained} = -2(logL_{Full,Constrained} - logL_{Restricted,Constrained}) \quad (10)$$

These likelihood ratio tests are then averaged across the *m* datasets using Equation 11.

$$\overline{LR}_{Constrained} = \frac{1}{m}\sum_{l=1}^{m} LR_{Constrained,l} \quad (11)$$

5. Calculate the average relative increase in variance (*k* is the number of degrees of freedom for the complete-data likelihood ratio test):

$$ARIV = \frac{m+1}{k(m-1)}(\overline{LR} - \overline{LR}_{Constrained}) \quad (12)$$

6. Calculate D_3:

$$D_3 = \frac{\overline{LR}_{Constrained}}{k(1 + ARIV)} \quad (13)$$

7. Calculate the denominator degrees of freedom, *v*.

$$\nu = 4 + (km - k - 4)\left[1 + (1 - \frac{2}{km - k})\frac{1}{ARIV}\right]^2 \quad (14)$$

8. Compare D_3 to an *F* distribution with numerator degrees of freedom equal to *k* (the number of degrees of freedom for the complete-data likelihood ratio test) and denominator degrees of freedom equal to *v*.

As one might guess from the above outline, implementing the D_3 test can be both difficult and time consuming. Because multiparameter significance testing is an essential part of multilevel analyses, the difficulty of implementing the D_3 test is a major barrier to using multilevel imputation in practice. To overcome this difficulty, the MMI_ANALYZE macro performs a pooled likelihood ratio test for two user-specified models.

THE MMI_ANALYZE MACRO

The macro is available at the following website: http://www.mistlerconsulting.com/software.html. Because the macro is very long, I recommend setting up the macro using the %INCLUDE function. Example 1 below shows the syntax for running MMI_ANALYZE with all parameters omitted.

Example 1:

```
%mmi_analyze(
    data                = ,
    cluster             = ,
    dv                  = ,
    full_fixed          = ,
    full_random         = ,
    restricted_fixed    = ,
    restricted_random   = ,
    ddfm                = ,
    type                = );
```

The DATA line specifies the name of the data set to be analyzed. CLUSTER is the name of the cluster variable, or level-2 ID variable. DV is the dependent variable for the analysis. FULL_FIXED names all of the independent variables to be included as fixed effects in the analysis (the intercept is included automatically, so should not be specified by the user). These are the variables that would be specified on the MODEL line of an analysis in PROC MIXED. FULL_RANDOM names all of the independent variables to be included as random effects (the intercept is included automatically, so should not be specified by the user). These are the effects that would be specified on the RANDOM line in PROC MIXED in addition to the intercept. The RESTRICTED_FIXED and RESTRICTED_RANDOM are used to analyze a second (restricted) model that is nested within the full model. These lines will be discussed further in the next section. For now, know that they can be left blank if you only wish to run a single model. DDFM specifies the method for computing denominator degrees of freedom in both models. The default for DDFM is SATTERTHWAITE, but any method available in PROC MIXED can be used (CONTAIN, BETWITHIN, RESIDUAL SATTERTHWAITE, or KENWARDROGER). Note, however, that the DDFM option changes the degrees of freedom method for the tests of fixed effects, but not for the pooled likelihood ratio test. The degrees of freedom calculation for the pooled likelihood ratio test is calculated separately. TYPE specifies the covariance structure for both models. The default for TYPE is UN (unstructured), but the VC (variance components) option can also be used. When the MMI_ANALYZE macro is called, it runs the requested analysis in PROC MIXED separately for each imputed data set. Also, the macro does not use the repeated line for PROC MIXED. As such, any analyses that require use of the repeated line cannot be performed with MMI_ANALYZE. The macro then pools the parameters and standard errors, and it also tests the significance of each fixed effect against an approximate *t* distribution.

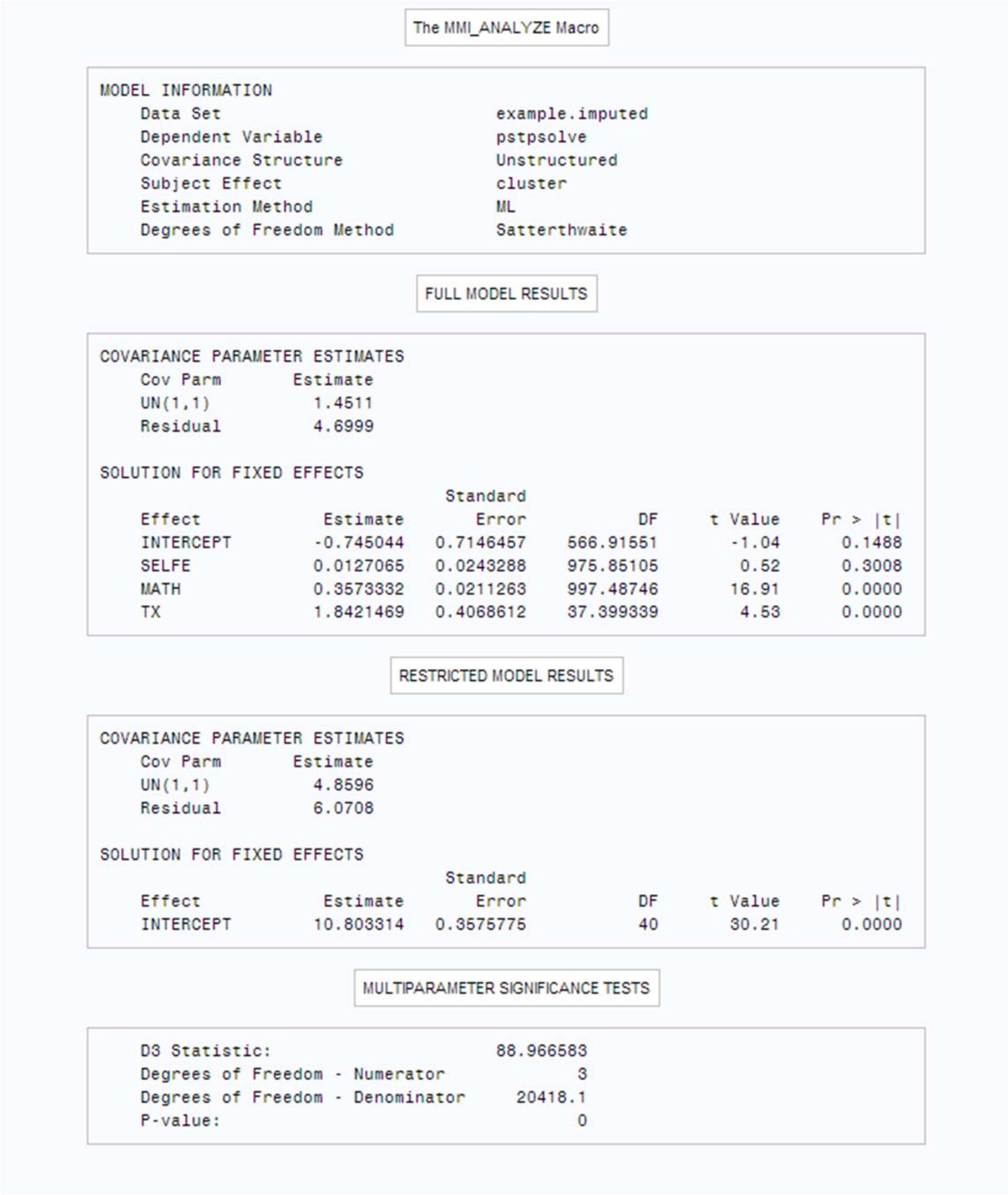
EXAMPLE

Recall that we wish to predict PSTPSOLVE from the fixed effects of MATH, SELFE, and TX (Equation 6). Our analysis can be performed using the syntax below.

Example 2:

```
%mmi_analyze(
    data                = example.imputed,
    cluster             = cluster,
    dv                  = pstpsolve,
    full_fixed          = selfe tx math,
    full_random         = ,
    restricted_fixed    = ,
    restricted_random   = ,
    ddfm                = sat,
    type                = un);
```

Our dependent variable, PSTPSOLVE, is entered on the DV line. All of our predictors are entered on the FULL_FIXED line because we wish to examine their fixed effects (the intercept is automatically included, so we don't have to specify it). We leave the FULL_RANDOM line blank because we don't believe that any of the predictors will have random effects (except for the intercept, which is automatically included as a random effect). We are not currently interested in analyzing a second model, so RESTRICTED_FIXED and RESTRICTED_RANDOM are left blank. The output of the macro is similar to that of PROC MIXED (Output 1). Note that the output for the restricted model refers to an intercept-only model and can be ignored if the intercept-only model is not of interest.



```
                              The MMI_ANALYZE Macro

MODEL INFORMATION
    Data Set                              example.imputed
    Dependent Variable                    pstpsolve
    Covariance Structure                  Unstructured
    Subject Effect                        cluster
    Estimation Method                     ML
    Degrees of Freedom Method             Satterthwaite

                               FULL MODEL RESULTS

COVARIANCE PARAMETER ESTIMATES
    Cov Parm       Estimate
    UN(1,1)          1.4511
    Residual         4.6999

SOLUTION FOR FIXED EFFECTS
                                   Standard
    Effect            Estimate        Error           DF      t Value     Pr > |t|
    INTERCEPT        -0.745044    0.7146457    566.91551        -1.04       0.1488
    SELFE            0.0127065    0.0243288    975.85105         0.52       0.3008
    MATH             0.3573332    0.0211263    997.48746        16.91       0.0000
    TX               1.8421469    0.4068612    37.399339         4.53       0.0000

                            RESTRICTED MODEL RESULTS

COVARIANCE PARAMETER ESTIMATES
    Cov Parm       Estimate
    UN(1,1)          4.8596
    Residual         6.0708

SOLUTION FOR FIXED EFFECTS
                                   Standard
    Effect            Estimate        Error           DF      t Value     Pr > |t|
    INTERCEPT        10.803314    0.3575775           40        30.21       0.0000

                        MULTIPARAMETER SIGNIFICANCE TESTS

    D3 Statistic:                         88.966583
    Degrees of Freedom - Numerator                3
    Degrees of Freedom - Denominator        20418.1
    P-value:                                      0
```

Output 1. Output from the MMI_ANALYZE macro

As seen in Example 2, MMI_ANALYZE always runs two models – a full model and a restricted model. If the RESTRICTED_FIXED and RESTRICTED_RANDOM lines are left blank, then the restricted model contains the intercept as both a fixed effect and a random effect, but nothing else (Equation 7). The output for the restricted model is printed separately, under “RESTRICTED MODEL RESULTS” (see Output 1). The output for the D_3 statistic contains the D_3 statistic, the numerator degrees of freedom, the denominator degrees of freedom, and the p-value from an approximate *F* distribution. To illustrate the D_3 statistic, let’s say we would like to examine the total predictive power of our treatment (TX) and math self-efficacy scores (SELFE) in a single test. To do this, we can simply compare our full model to a model without TX or SELFE. We know that the two models are nested, because we can substitute a value of zero for the coefficients of SELFE and TX (β_2 and β_3, respectively) from Equation 6 to arrive at the restricted model below.

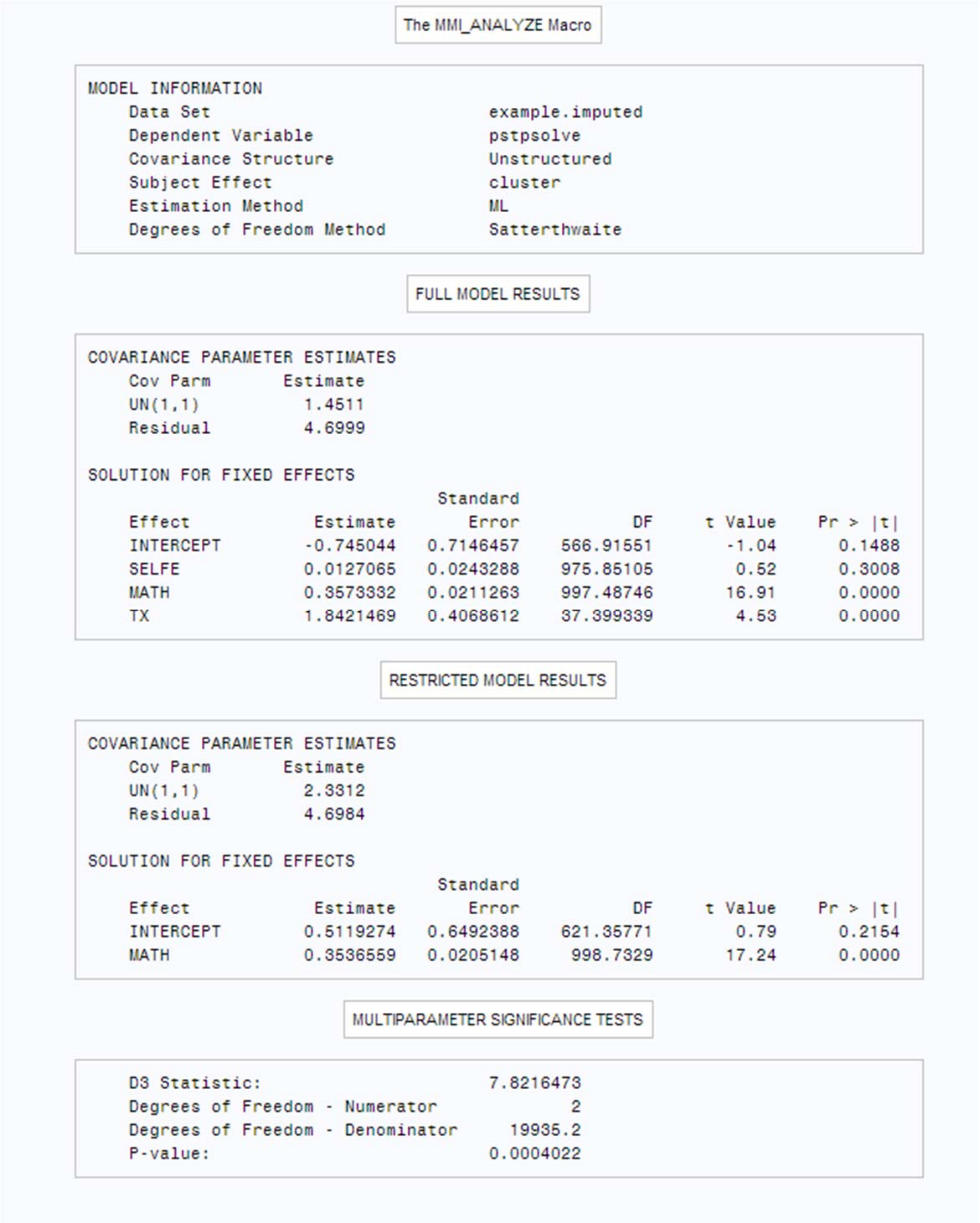
$$\mathrm{PSTPSOLVE}_{ij} = \beta_0 + \beta_1\left(\mathrm{MATH}_{ij}\right) + b_{0j} + \varepsilon_{ij} \qquad (16)$$

Both models can be analyzed with the syntax below.

Example 3:

```
%mmi_analyze(
    data                = example.imputed,
    cluster             = cluster,
    dv                  = pstpsolve,
    full_fixed          = selfe tx math,
    full_random         = ,
    restricted_fixed    = math,
    restricted_random   = ,
    ddfm                = sat,
    type                = un);
```

Notice that the parameters for RESTRICTED_FIXED are the same as for FULL_FIXED, but without TX or SELFE. The output for this analysis is shown in Output 2. Notice that the results for the full model in Example 3 are identical to the results for the full model in Example 2. This is as it should be, since we did not change the full model. However, the results for the restricted model are quite different, as are the results for the D_3 statistic.



```
The MMI_ANALYZE Macro

MODEL INFORMATION
    Data Set                             example.imputed
    Dependent Variable                   pstpsolve
    Covariance Structure                 Unstructured
    Subject Effect                       cluster
    Estimation Method                    ML
    Degrees of Freedom Method            Satterthwaite

FULL MODEL RESULTS

COVARIANCE PARAMETER ESTIMATES
    Cov Parm        Estimate
    UN(1,1)           1.4511
    Residual          4.6999

SOLUTION FOR FIXED EFFECTS
                                     Standard
    Effect             Estimate         Error           DF     t Value    Pr > |t|
    INTERCEPT         -0.745044     0.7146457    566.91551       -1.04      0.1488
    SELFE             0.0127065     0.0243288    975.85105        0.52      0.3008
    MATH              0.3573332     0.0211263    997.48746       16.91      0.0000
    TX                1.8421469     0.4068612    37.399339        4.53      0.0000

RESTRICTED MODEL RESULTS

COVARIANCE PARAMETER ESTIMATES
    Cov Parm        Estimate
    UN(1,1)           2.3312
    Residual          4.6984

SOLUTION FOR FIXED EFFECTS
                                     Standard
    Effect             Estimate         Error           DF     t Value    Pr > |t|
    INTERCEPT         0.5119274     0.6492388    621.35771        0.79      0.2154
    MATH              0.3536559     0.0205148     998.7329       17.24      0.0000

MULTIPARAMETER SIGNIFICANCE TESTS

    D3 Statistic:                        7.8216473
    Degrees of Freedom - Numerator               2
    Degrees of Freedom - Denominator       19935.2
    P-value:                             0.0004022
```

Output 2. Output from the MMI_ANALYZE macro

As seen in Output 2, the D_3 statistic is significant, $F(2,5150.9) = 8.68$, $p < .001$. Consistent with the interpretation of a likelihood ratio chi-square test from a complete-data analysis, the approximate F statistic indicates that, considered as a set, the treatment variable and the math self-efficacy scores account for a significant improvement in model fit.

SINGLE-LEVEL REGRESSION

The MMI_ANALYZE macro is also able to pool estimates from and to perform the pooled likelihood ratio test on single-level regression models that can be run in PROC MIXED. To analyze a set of single-level models, specify only the DATA, DV, FULL_FIXED, and RESTRICTED_FIXED parameters as shown in Example 4.

Example 4:

```
   %mmi_analyze(
       data                = example.imputed,
       dv                  = pstpsolve_avg,
       full_fixed          = tx selfe_avg math_avg,
       restricted_fixed    = math_avg);
```

In Example 4, I performed a regression using the intervention code, TX, and the cluster-averages of the level-1 variables. Example 4 estimates two models. The full model is:

$$\mathrm{PSTPSOLVE\_AVG}_j = \beta_0 + \beta_1(\mathrm{MATH\_AVG}_j) + \beta_2(\mathrm{SELFE\_AVG}_j) + \beta_3(\mathrm{TX}_j) + \varepsilon_j \qquad (17)$$

where PSTPSOLVE_AVG$_j$ is the average of PSTPSOLVE for cluster *j*, MATH_AVG is the average of MATH for cluster *j*, and SELFE_AVG is the average of SELFE in cluster *j*. β_0 is the intercept. β_1, β_2, and β_3 are slope coefficients, and ε_j is the residual for cluster *j*. The restricted model from Example 4 is:

$$\mathrm{PSTPSOLVE\_AVG}_j = \beta_0 + \beta_1(\mathrm{MATH\_AVG}_j) + \varepsilon_j \qquad (18)$$

We know that the two models are nested, because we can substitute a value of zero for the coefficients of SELFE_AVG and TX_AVG (β_2 and β_3, respectively) from Equation 17 to arrive at the restricted model in equation 18. For single-level regression, MMI_ANALYZE sets DDFM to RESIDUAL.

CONCLUSION

Likelihood ratio tests for multiply imputed multilevel data are difficult to implement. Given the prevalence of multilevel data, the lack of software for performing likelihood ratio tests on multiply imputed data is problematic. I developed a SAS macro, MMI_ANALYZE, which computes a complex pooled likelihood ratio statistic from multiply imputed data. This paper demonstrates the use of this new tool in the context of a two-level multilevel model.

The macro mentioned in this paper is available at: http://www.mistlerconsulting.com/software.html. I plan to update the macro regularly to increase user-friendliness and add greater functionality. I welcome any comments and suggestions that you may have for updates to the macro.

REFERENCES

- Meng, X-L., & Rubin, D. B. (1992). Performing likelihood ratio tests with multiply-imputed data sets. *Biometrika*, 79(1), 103-111.
- Mistler, S.A. (2013). *A SAS macro for applying multiple imputation to multilevel data*, in Proceedings of the SAS Global Forum 2013, San Francisco, California: Contributed Paper (Statistics and Data Analysis) 438-2013.
- Montague, M., Enders, C., & Dietz, S. (2011). Effects of cognitive strategy instruction on math problem solving of middle school students with learning disabilities. *Learning Disability Quarterly*, 34(4), 262-272.
- Raudenbush, S.W., & Bryk, A.S. (2002). *Hierarchical linear models: Applications and data analysis methods* (2nd ed. Vol. 1). Thousand Oaks, CA: Sage Publications.
- Rubin, D. B. (1987). *Multiple imputation for nonresponse in surveys*. Hoboken, NJ: Wiley.
- Schafer, J. L. (1997). *Analysis of incomplete multivariate data*. Boca Raton, FL: Chapman & Hall.
- Singer, J.D., & Willett, J.B. (2003). *Applied longitudinal data analysis: Modeling change and event occurrence*. Oxford, NY: Oxford University Press.

ACKNOWLEDGMENTS

I would like to thank my SAS mentor, David Pasta, for his careful review and valuable comments. I would also like to thank my advisor, Dr. Craig Enders, for his insights into missing data theory.

CONTACT INFORMATION

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

Stephen Mistler
E-mail: stephen.mistler@asu.edu
Web: www.mistlerconsulting.com