Meta-Analysis using SAS

L. V. Madden and P. A. Paul
Ohio State University

This a supplemental (*e-X*tra) file to be read in conjunction with:

> Madden, L. V., and Paul, P. A. Meta-analysis for evidence synthesis in plant pathology: An overview. Phytopathology.

Meta-analysis can be done using specialized macros or programs, or with mixed-model procedures such as MIXED in SAS. The utilization of these procedures requires the use of some unusual combination of options, so the approach is not obvious. van Houwelingen et al. (2002; Stat Med. 21:589-624) is a good introduction for MIXED users. Below, some even easier ways of doing the identical analysis are presented.

With MIXED, one can perform maximum-likelihood (ML) or restricted maximum-likelihood (REML) model fitting. Many authors like to use ML in meta-analysis, but both methods are fine. If ML is used, then some other options should be chosen for degrees of freedom (df) to be consistent with the large-sample perspective being considered. ML and REML will give slightly different results. For small number of studies (<30), REML would be better.

The code below is prepared for instructional purposes. There are no warranties. The data set consists of simulated data, with K=50 studies (these are not the real data used in the above listed paper). The following variables are in the data set:

efft:	effect size per study;
var_efft:	variance of the effect size per study (square of SE of effect size) (known as sampling variance, derived from residual mean square)
moder1:	a categorical moderator variable (e.g., cultivars 1, 2, 3);
moder2:	a continuous moderator variable (e.g., avg. T).
studyno:	label for the study (1,…,50)

Data are read into SAS file demo using the following code:

```
data demo;                                   *<--simulated data;
input moder1 moder2 efft var_efft studyno;
wgt = 1/var_efft;                            *<--need weight per study (=1/var);
datalines;
1     21    1.13286     20.1732     1
0     19    0.91970     1.4405      2
1     22    0.72507     5.8174      3
1     24    -0.47340    24.8908     4
1     24    0.56236     2.4292      5
2     21    3.43668     6.4030      6
1     15    1.60458     2.1051      7
0     15    1.42985     1.6785      8
0     18    1.58239     1.5939      9
0     20    1.92021     0.0963      10
1     21    1.95475     3.0475      11
1     18    1.18504     0.4636      12
1     22    1.16712     1.1695      13
0     18    0.70154     1.3622      14
0     25    0.67354     0.8050      15
0     19    0.99711     2.0288      16
```

```
0      23     2.27181      5.1034       17
2      22     2.12609      23.5246      18
2      21     2.76217      1.9400       19
2      18     1.36925      0.7100       20
2      17     0.46791      3.3591       21
1      20     0.86908      4.0169       22
2      21     1.45729      18.6656      23
2      18     2.64130      1.8637       24
0      19     1.76658      0.3448       25
0      15     0.69369      4.2905       26
0      18     1.61441      0.0928       27
2      23     2.63756      22.3204      28
0      17     1.50678      0.2252       29
0      21     -0.11565     0.1711       30
0      15     0.05201      0.0692       31
2      22     -1.39402     3.9929       32
2      23     1.35610      1.0142       33
1      21     1.10818      5.0666       34
0      15     0.22629      0.9755       35
2      16     -0.03458     15.4405      36
0      20     1.87402      1.0009       37
1      19     2.54478      20.9345      38
1      19     1.28713      0.4589       39
1      16     0.22484      3.7634       40
0      21     3.02889      1.6573       41
2      20     0.55123      2.6158       42
0      22     -0.74106     0.2730       43
2      19     1.71147      1.2556       44
2      18     0.43599      9.7652       45
1      20     1.20071      0.9767       46
2      25     2.06417      6.0231       47
0      18     1.35886      0.6265       48
1      26     0.75691      12.5355      49
0      19     0.96668      0.3826       50
;
run;
```

Random-effects analysis. Below is the code for the MIXED procedure to perform a ML-based random-effects meta-analysis.

```
title 'Demo of univariate meta-analysis with PROC MIXED in SAS';
title2 'ML analysis in SAS, RANDOM effects, no moderators';
proc mixed data=demo method=ml covtest;
class studyno;              *<--indicate that study is a class/factor variable;
weight wgt;                 *<--weight is inverse of sampling variance per study;
model efft = / chisq s df=10000; *<--model gives just the global mean (intercept);
random studyno;             *<--define a random study effect (get variance);
repeated;                   *<--this and next line force fixed sampling variances;
parms (1) (1) / eqcons=2;  *<--forces the sampling-variance to be 1*(1/weight);
estimate 'meanES' int 1
       / cl df=10000;       *<--another way of displaying est. expected effect size;
run;
```

Note: In above, a 'trick' is used to fix the so-called (within-study) sampling variances at pre-determined values. This is done by forcing the residual to be '1' for all studies, *and* to simultaneously fix the WEIGHT (which is really a within-study weight) as 1/(sampling variance). Here, the weight is called `wgt`. In the

PARMS statement, there are two terms: the first one is the initial guess of the among-study variance, and the second is the initial (and final) choice of the within-study sampling variance (1), which is held constant with the eqcons=2 option (i.e., not allowing the second parameter to vary).

In the output, the among-study variance estimate is given in the table of Covariance Parameter Estimates (studyno). The estimated variance here is 0.3272. The estimated standard error of the estimated among-study variance is 0.1587. The same table gives an 'estimate' of 1 for the Residual variance. But, because there are weights (that vary with study), this really means that there is a *different fixed* sampling variance for each study [=1*(1/weight)].

Covariance Parameter Estimates

Cov Parm	Estimate	Standard Error	Z Value	Pr > Z
studyno	0.3272	0.1587	2.06	0.0196
Residual	1.0000	0	.	.

Fit Statistics

-2 Log Likelihood	165.2
AIC (smaller is better)	169.2
AICC (smaller is better)	169.4
BIC (smaller is better)	173.0

PARMS Model Likelihood Ratio Test

DF	Chi-Square	Pr > ChiSq
1	6.09	0.0136

Solution for Fixed Effects

Effect	Estimate	Standard Error	DF	t Value	Pr > |t|
Intercept	1.0913	0.1731	49	6.30	<.0001

In the Solution for Fixed Effects, the “Intercept” (1.0913) is the estimated expected (mean) effect size. The estimated standard error of the estimated expected effect size (0.1731) is also given. The ESTIMATE statement in the MIXED code gives another way of displaying the same estimated effect size (see below). In this output, the very large df value is used (1E4 = 10000), which means that the t value (estimate effect size divided by standard error) can be considered a standard normal value. The Pr value is the *P* value for the standard normal test. With a default alpha of 0.05, the Lower and Upper values (below) give the limits of a 95% confidence interval.

Estimates

Label	Estimate	Standard Error	DF	t Value	Pr > |t|	Alpha	Lower	Upper
meanES	1.0913	0.1731	1E4	6.30	<.0001	0.05	0.7519	1.4306

Next, a more difficult way to do the same analysis is shown, which is often given in meta-papers. First create a new file.

```
data var1; set demo; *<--make new file;
est = var_efft; *<--define sampling variances as estimates of the residual variances;
keep est studyno;
data var2;
est = 1;                    *<--define the starting guess for the among-study variance;

data variance; set var2 var1; *<--stack the variance files (among-study, then all
the within-study sampling variances;
run;
```

In above, a special (PARMS DATA: PDATA) file was made, which starts with an arbitrary first guess at the among-study variance, followed by the 50 (here) sampling variances (one from each study). It is a good idea to have the original data file sorted by study (studyno here) before doing any of this. It is also a good idea to printout this file, to make sure you created it in the right way.

```
title2 'a look at the guess of among-study var., and all the sampling variances';
title3 '(the sampling variances will not change in the model fitting)';
proc print data=variance;
run;
```

The output is not displayed here, but it would show 51 records. The first one would be a 1 for “est” variable, and the next 50 would simply be the original sampling variances (read into the demo file previously). Now, the data are analyzed with the following code.

```
title2 'ML analysis in SAS, RANDOM effects (use of PDATA= for sampling variances)';
proc mixed data=demo method=ml covtest;
class studyno;
model efft = / chisq s df=10000;        *<--note: *no* weight;
random studyno;
repeated / group=studyno type=simple;
parms / pdata=variance eqcons=2 to 51;  *<--PDATA= gives the variances;
                                        *^--All but the first are held constant;
estimate 'meanES' int 1 / cl df=10000;
run;
```

In the output for the PDATA approach (not shown), the table of Covariance Parameter Estimates first displays the among-study variance estimate (.3272), identical to the result with the earlier MIXED run (using the WEIGHTS and =1 for residual), followed by the 50 fixed sampling variances. These are the same as those listed in the PDATA file, and read into the original demo file. The model fit results (meanES, standard error, likelihood, confidence interval, etc.) are identical (to the results obtained with the easier-to-use WEIGHTS approach.

There is a third way to do the same analysis. This uses the PARMS statement in yet another way. One writes out (manually) the sampling variances for each study, after first listing the initial guess for among-study variance. Example:

```
parms (1) (20.1732) (1.4405) (5.8174) ... (0.3826) / eqcons 2 to 51;
```

The option (after the /) says that variance parameters 2 through 51 are held constant. Other lines of code are unchanged. The data should be sorted by study (`studyno`), and the parameters should be listed in the same order as the studies in the data file. As you can see, this approach becomes extremely tedious for more than a few studies. The take-home message: the first approach to using MIXED (given above) is by far the easiest for univariate analyses, and this approach is identical to the other more tedious approaches.

Fixed-effects analysis. Now we go back to the first model-fitting approach (using the combination of WEIGHT and fixing the [single] residual variance at 1 [which really gives fixed separate sampling variances for different studies]). One can perform a fixed-effect (common effect) analysis using the following code.

```
title2 'ML analysis in SAS, FIXED (common) effects (no among-study variance)';
proc mixed data=demo method=ml covtest;
class studyno;
weight wgt;
model efft = / chisq s df=10000;
repeated;                              *<--note: no random effects;
parms (1) / eqcons=1;                  *<--forces sampling-variance to be 1*(1/weight) ;
estimate 'meanES' int 1 / cl df=10000;
run;
```

Note that there is only one parameter in this model, and thus the PARMS statement is simpler than for the other model fits. The output (not shown) indicates that meanES (the estimated common effect size) is 1.0047, and its estimated standard error is 0.1129, both of which are different from the values obtained with the random-effects analysis. The R^2 statistic of Higgins & Thompson (not a coefficient of determination) is obtained by dividing the SE of the effect size for the random-effects model by the SE of the effect size for fixed-effects model, and squaring this. For the example: $R^2 = (0.1731/0.1129)^2 = 2.3$.

Random-effects and moderator variable. Next we show how to determine the effect of a categorical moderator variable (`moder1`; here with three levels, such as three cultivars) on the effect size, with a random-effects analysis.

```
title2 'ML analysis, RANDOM effects of study, CLASS moderator';
proc mixed data=demo method=ml covtest;
class studyno moder1;                  *<--list moder1 as a class/factor variable;
weight wgt;
model efft = moder1 / chisq s df=10000; *<--moder1 gets listed in model statement;
random studyno;
repeated;
parms (1) (1) / eqcons=2;
lsmeans moder1 / cl df=10000;          *<--get means for each group;
run;
```

A portion of the output is shown below. Type 3 Tests of Fixed Effects simply are tests for the effect of the moderator variable (in this case). This table does not exist in the analysis without moderator variables. With a forced large denominator df (1E4 = 10000), the F test and the chi-square test (both shown) give identical P values. The numerator df is 2 (=3-1). Note: divide the chi-square value by the numerator df (2 here) and one gets the F value. The estimated expected values for all levels of moder1 are given because of the LSMEANS statement. Confidence intervals are also shown; with the large df, the t value is the same as a standard normal value.

Type 3 Tests of Fixed Effects

Effect	Num DF	Den DF	Chi-Square	F Value	Pr > ChiSq	Pr > F
moder1	2	1E4	1.04	0.52	0.5951	0.5951

Least Squares Means

Effect	moder1	Estimate	Standard Error	DF	t Value	Pr > |t|	Alpha	Lower	Upper
moder1	0	0.9874	0.2074	1E4	4.76	<.0001	0.05	0.5808	1.3939
moder1	1	1.1644	0.4029	1E4	2.89	0.0039	0.05	0.3747	1.9541
moder1	2	1.4905	0.4596	1E4	3.24	0.0012	0.05	0.5895	2.3914

Now we analyze a continuous moderator variable (`moder2`).

```
title2 'ML analysis, RANDOM effects, CONTINUOUS moderator';
proc mixed data=demo method=ml covtest;
class studyno ;                          *<--moder2 is NOT given in class statement;
weight wgt;
model efft = moder2 / chisq s df=10000; *<--moder2 in model statement;
random studyno;
repeated;
parms (1) (1) / eqcons=2 ;
run;
```

There is no LSMEANS statement because moder2 is continuous. The Type 3 Test output is also relevant for continuous variables, but the results are not shown here. The other important output here is the Solution for Fixed Effects. As shown below, this gives the estimated intercept (1.1187) and slope (-0.00144) for the relation between the effect size and moder2.

Solution for Fixed Effects

Effect	Estimate	Standard Error	DF	t Value	Pr > |t|
Intercept	1.1187	1.3608	48	0.82	0.4151
moder2	-0.00144	0.07073	1E4	-0.02	0.9838

For analysis of moderator variables, it remains important to report the among-study variance estimate. These were not given here, but SAS displays these in the same manner as given for no moderator variables.

In all of the above analyses, one can obtain REML-based results simply by changing the MIXED statement to `method=reml`. One would also take out the `df=` or `ddf=` options.

Indices of heterogeneity. For the most part, we strongly advocate ML or REML fitting of random-effect or mixed-effect models to estimated effect sizes from the different studies. However, it must be pointed out that moment-based methods remain very popular, especially for univariate meta-analyses. A hallmark of the

moment-based approach is the Q statistic of Cochran. This is really a weighted sum of squares about the estimated expected value, where the latter is based on a FIXED-effects analysis. Q is not obtained in a likelihood-based analysis. One can obtain Q from a fixed-effects analysis using the GLM procedure (and weights). Almost all the output from GLM is incorrect, however, because one cannot fix the residual variances for each study at the required values. The only relevant and correct piece of output from GLM (for this type of application) is the Error Sum of Squares, which is the same as Q (because of the weights). The following code obtains Q from a run of GLM (based on weighted least squares), but only displays a subset of the results. Ignore everything but this one statistic. Then, the data step calculates the P value for the chi-squared test of homogeneity (Pq), and also calculates two indices of Higgins and Thompson for the degree of heterogeneity.

```
proc glm data=demo;
title2 "Cochran's Q, df, test of homogeneity (Prob), and Higgins-Thompson H2 & I2";
ods listing select overallanova;        *<--look only at a table that contains SSE;
ods output overallanova=q_out;          *<--store Q (=SSE) for manipulation below;
class studyno;
weight wgt;
model efft = / solution;
run;
data q_out2; set q_out(where=(Source='Error'));      *<--manipulate the Q value;
Q = ss;
H2 = Q/df;                          *<--H-squared statistic of Higgins and Thompson;
I2 = 100*(H2-1)/H2;                 *<--I-squared statistic of Higgins and Thompson;
Pq = 1 - probchi(Q,df);             *<--P value for test of homogeneity (Cochran Q test);
keep Q  df Pq H2 I2 I22;

proc print data=q_out2;
       var Q df Pq H2 I2 ;
run;
```

The output below shows the relevant statistics.

Obs	Q	DF	Pq	H2	I2
1	60.0238	49	0.13441	1.22498	18.3658

The P value is 0.134; this implies that the among-study variance is not significantly different from 0. But, this test is not very powerful, and is not overly useful. The moment-based among-study variance can be estimated based on the above q_out2 file and summations of weights and squared weights in the demo file. Details are in the Appendix of Madden and Paul. The code is given here:

```
data demo2; set demo;                   *<--temporary file;
wgt2=wgt**2;                            *<--get weights squared (same as sampling SD
                                              to -4th power);
proc means data=demo2 sum noprint;      *<--get and store the sum of weights and weights-
squared;
var wgt wgt2;
output out=demo2out sum(wgt)=wgts sum(wgt2)=wgt2s;
run;
data q_out3; merge q_out2 demo2out;     *<--combine/merge files (Q, etc., with sum
                                              of weights);
data q_out3; set q_out3;                *<--get c coefficient (see Appendix in
                                              Madden & Paul);

c = wgts - (wgt2s/wgts);
variance_MM = (Q - df)/c;               *<--get moment-based among-study
                                              variance estimate;
```

```
title2 'Moment-based estimate of the among-study variance';
proc print data=q_out3;
var Q c variance_MM;
run;
```

Results are not shown here, but the estimated variance is 0.15; this is about half of the ML-based variance estimate (0.33). In general, we prefer ML or REML analyses.

Conclusions. For more information, and more examples, on the use of SAS for meta-analysis, please contact the authors (madden.1@osu.edu or paul.669@osu.edu) or consult the first author's website for access to an additional and more extensive demonstration program.