

Meta-Analysis and Matrix Multiplication: Adapting an IML-Based Macro for DerSimonian and Laird's Error Calculation to use Base SAS®

Venita DePuy, Bowden Analytics

ABSTRACT

Proc IML® is a wonderful tool; but not all companies opt to purchase the SAS license for that product. This paper summarizes my journey to adapt Hertzmark & Spiegelman's %METAANAL macro, which produces DerSimonian-Laird estimators for meta-analysis, to use Base SAS. The macro itself is also provided.

INTRODUCTION

The original paper by DerSimonian & Laird presents a random effects approach to combine results from multiple clinical trials comparing the two treatments. This estimator was actually previously proposed by Cochran, but is most commonly known as DerSimonian and Laird's Q test, per Takkouche et al's 1999 paper. That paper also showed that asymptotically, the DerSimonian-Laird Q statistic gives the correct Type I error under the null hypothesis, and is preferable to other options for validity, power, and computational ease.

OVERVIEW OF META-ANALYSES

Meta-analysis is basically method of combining the results from multiple independent studies, to obtain a composite result from all available data. Due to the differences between studies – whether in inclusion/exclusion criteria, methodology, or other aspects – the subjects should not simply be all lumped together and the data analyzed as if it were a single study. Instead, after studies are reviewed and considered to be “combinable” from an analytic standpoint, they should be combined using appropriate methods.

One of the most common methods to perform meta-analysis is the DerSimonian-Laird estimator, based on their 1986 paper.

OVERVIEW OF THE ORIGINAL DERSIMONIAN-LAIRD PAPER

The original DerSimonian-Laird paper reviewed 8 published meta-analyses and discussed the lack of consistency in estimating the homogeneity of treatment effect before pooling. Then, they proposed their random effects approach (the DerSimonian-Laird estimator) to combine the results from multiple studies while incorporating the heterogeneity of effects in the overall analysis of treatment efficacy.

In simpler terms – a fixed effects model (like the standard least-squares regression) assumes a constant treatment effect between subjects. A random effects model instead assumes that the treatment effect (if the treatment effect is considered random) for each subject is like sampling from a normal distribution of treatment effects. The DerSimonian-Laird estimator assumes that the treatment effect found in each study is taken from a distribution of treatment effects.

WHEN ARE STUDIES “COMBINABLE”?

There isn't a hard and fast checklist of whether or not studies can be combined. Ideally, you want to use studies that are as similar as possible. In reality, there will probably be a limited number of studies available for your area of interest, and an even smaller number of studies to which you have access to sufficient information.

In my personal experience, the best option is to display the results of individual studies in combination with the results from the meta-analysis, and allow the reader to see how similar (or dissimilar) the results may be across studies. For instance, assume you have 4 studies. You can do a forest plot of the treatment effect (and confidence interval) that displays the meta-analysis at the top, followed by each of

the 4 studies. You can design summary tables with 1 column for each study plus a column for the meta-analysis. If data comes from disparate sources or studies (for example, you have data from both observational databases and clinical trials), I would also suggest presenting results within those (meta-analysis of observational databases, meta-analysis of clinical trials, and overall meta-analysis) for transparency.

Overall, it is important to include all relevant trials, even (especially?) if the results are contradictory.

I also recommend summarizing the studies, and their differences. Was one conducted much earlier than the others, and standards of care may have shifted? Is another conducted in a different country, that may have different health care / hospitalization practices? Perhaps a third study had more rigid inclusion criteria that limited subjects to those who were overall quite healthy, and had different results from a fourth study with broader inclusion criteria that ended up enrolling a lot of less healthy subjects. How much did the dose level, or treatment duration, vary between studies?

It's nice to have consistent results between studies. My personal opinion is that, the more disparate the results between studies, the more explanation of differences & similarities between studies is needed.

SO HOW DO I ACTUALLY DO THIS?

First, some notation, based on DerSimonian & Laird's paper. We'll assume we have k independent studies; each of them has n_T subjects in the treatment arm and n_C subjects in the control arm. A proportion of subjects in each arm experience some event, r_T and r_C . We assume that the number of subjects experiencing the event in each study are independent binomial random variables with probabilities p_{Ti} and p_{Ci} , where $i = 1$ to k . Three measures discussed in DS&L are the risk difference $p_{Ti} - p_{Ci}$, the relative risk p_{Ti} / p_{Ci} , and the relative odds $[p_{Ti} / (1 - p_{Ti})] / [p_{Ci} / (1 - p_{Ci})]$.

Each of the k studies has observed treatment effect $y_i = \theta_i + e_i$, where θ_i is the true treatment effect and e_i is the sampling error for the i th study. The variance of e_i is the sample variance s_i^2 . (When y_i is the difference in proportions $r_{Ti} - r_{Ci}$, we can estimate $s_i^2 = r_{Ti}(1 - r_{Ti}) / n_{Ti} + r_{Ci}(1 - r_{Ci}) / n_{Ci}$.) We assume that the true treatment effect in the i th study, θ_i , is $\theta_i = \mu + \delta_i$, where μ is the mean of the distribution of possible treatment effects and δ_i is the deviation of the i th study's effect from that population mean, where $\text{var}(\delta) = \Delta^2$.

We use the statistic Q , the sum of squares of the treatment effect about the mean, to evaluate the homogeneity of treatment effect. Where y_i is the observed treatment effect from the i th study and w_i is typically the inverse of the i th sampling variance ($w_i = 1/s_i^2$), we have:

$$\bar{y}_w = \frac{\sum w_i y_i}{\sum w_i} \text{ and } Q = \sum w_i (y_i - \bar{y}_w)^2$$

where $\bar{y}_w$ is the weighted estimator of the treatment effect, and $Q \sim \chi^2_{k-1}$ under the null hypothesis of $H_0: \Delta^2 = 0$, assuming a sufficiently large sample size.

The Q statistic for difference in proportions is calculated as above and named Q_w . On the relative odds scale, the Q statistic (called Q_L) can also be derived, where $y_i = \text{LN} [r_{Ti} (1 - r_{Ci}) / r_{Ci} (1 - r_{Ti})]$, w_i and $\bar{y}_w$ are calculated as above, Q_L is calculated as per the Q equation above, and

$$s_i^2 = \frac{1}{n_{Ti} r_{Ti} (1 - r_{Ci})} + \frac{1}{n_{Ci} r_{Ci} (1 - r_{Ti})}$$

Further, when $\Delta^2 \neq 0$, Q_w is used to estimate Δ^2 where Q_w , w_i and $\bar{y}_w$ are as above and

$$\Delta_w^2 = \max\left\{0, \frac{Q_w - (k - 1)}{\sum w_i - \left(\frac{\sum w_i^2}{\sum w_i}\right)}\right\}$$

then the weighted least squares or Cochran's semi-weighted estimator of μ is

$$\mu_w = \frac{\sum w_i^* y_i}{\sum w_i^*} \text{ where } w_i^* = \frac{1}{1/w_i + \Delta_w^2} \text{ and } se(\mu_w) = \sqrt{\sum w_i^*}$$

where $se(\mu_w)$ is the asymptotic standard error of μ_w .

DerSimonian & Laird also discusses the method of Gilbert et al which assumes a common sampling variance s^2 , which results in a simpler calculation for Q , which we call Q_U . The calculation for Δ_w^2 is also simplified.

TO SUMMARIZE THE NOTATION...

Equations are provided above for different variations of the Q statistic; Q_w for the difference, Q_L for the log odds scale, and Q_U for the differences when variances are assumed to be the same between studies. It is important to note that DerSimonian & Laird found differing results for the test of homogeneity between Q_w , Q_L , and Q_U , so the choice of Q statistic should be determined *a priori*, if the test for homogeneity is the goal of the analysis.

CONFIDENCE INTERVALS

While DerSimonian & Laird does not explicitly talk about using their method to create confidence intervals (CIs) for treatment effects, the same calculations can be used to create confidence intervals, treating study as a random effect. Hertzmark & Spiegelman provided the calculations in their macro.

AND A FEW OTHER ESTIMATORS...

The Hertzmark & Spiegelman macro was updated in by Polyna Khudyakov in 2015 to include additional estimators for quantifying heterogeneity that were introduced in the 2013 paper by Takkouche, Khudyakov, Costa-Bouzas and Spiegelman. I have included those estimators in my version of their macro; please refer to that paper for discussion of the R_I , CV_B , and other estimators.

WHAT YOU NEED TO SPECIFY FOR THE MACRO

These are the macro parameters that are read in to the macro call (of both the original and my revised macro). They are described in more detail in Hertzmark & Spiegelman's paper.

Parameters:

- DATA: name of the dataset with coefficients and variances / standard errors
- STUDYLAB: name of variable containing names of different studies
- EXPLABEL: descriptive title of exposure
- OUTCOMELABEL: descriptive title of outcome
- COEFF: Values are T or F. T = results from initial analyses (which are read into this macro) as coefficients & standard errors. F = results provided as 95% confidence intervals

Required if COEFF=T:

- o BETA: name of variable containing coefficients
- o SE_OR_V: indicates whether you are providing standard errors (SE_OR_V = S) or providing the variances (SE_OR_V = V) of the coefficients
- o SE: name of the variable providing standard errors of coefficients (if SE_OR_V = S)
- o VAR: name of the variable providing variances of coefficients (if SE_OR_V = V)

Required if COEFF=F:

- o VARNAMELCB: name of variable containing lower 95% confidence bound
- o VARNAMEUCB: name of variable containing upper 95% confidence bound

- LOGLINEAR: T if study results entered into the macro are from log-linear models such as logistic, proportional hazards, log-binomial, or Poisson), F if results are instead from linear models. (If LOGLINEAR = T, macro will display the pooled odds ratio (OR) or risk ratio (RR) and its 95% CI)
- PRINTCOEFF (if LOGLINEAR=T): T = print coefficients in addition to RR/ORs, F = do not print the coefficients, just the RR/ORs
- POOLTYPE: method of pooling (random, fixed, or both) – defaults to “random” (recommended)
- WT: difference/change in exposure for which to calculate odds ratios – defaults to 1
- OUTDAT: name of output dataset. No dataset will be created if left blank.
- NOPRINT: F if output should be printed, T if output should not be printed
- PVALUEFORMAT: preferred numeric format of p-values – defaults to pvalue6.4

WALKING THROUGH THE MACRO

The macro itself is provided in the attached appendix.

The initial portion of the macro provides error checking to ensure that sufficient information is provided (either NOPRINT=F or OUTDAT not blank, for example). Then, initial calculations of beta coefficients (where COEFF=F), standard errors or variances (depending on values of COEFF and SE_OR_V), and odds ratio/risk ratio and its upper and lower bounds. The numbers of non-missing standard errors, variances, and betas are read into macro variables to check that all values are not missing.

After that point, the macros diverge; the original macro uses Proc IML, and I have created the equivalent calculations (although less concise) using data steps and similar components of Base SAS. I included comments throughout indicating what the original IML code was, for easy comparison.

First, the values of the betas, the labels of the individual studies, and the variances of the betas are read into macro variables (in lieu of an IML matrix). The weighted odds ratio (variable named ODR as previously, for consistency with the original macro) and its 95% confidence bounds (named LL & UU) are calculated. The inverse of the variance, W , is calculated and summed across studies, then W_{FIXED} , the weight in the fixed effects model, is calculated as W divided by the sum of W . The beta coefficients, multiplied by their weights in the fixed model, are summed across studies to calculate the pooled coefficients from the fixed effects model, $BETAF$. Its confidence bounds, $BLOWF$ and $BHIGHF$, are calculated using the standard error, which is the square root of the inverse of the sum of W . The pooled OR/RR for the fixed model, incorporating the $\&WT$ weight, are calculated along with its confidence bounds (ORF , $ORLOWF$, $ORHIGHF$ respectively). The Z-score is then calculated, squared to obtain the X^2 statistic, and the p-value ($PCHI2F$) calculated to determine whether the pooled beta coefficient, from the fixed model, is significantly different from zero.

The Q statistic is then calculated as the sum of the squared differences between the individual betas and the pooled betas, multiplied by the inverse of the variance (W). Note that this is Q_w per the DerSimonian & Laird paper. $PHET$, the p-value for the test of heterogeneity, is calculated from Q based on the X^2 distribution. The sum of the squared W s is also calculated. $TAU2$, the estimate of between-studies variance, is calculated as described for Δ_w^2 above.

$WRAND$, the weight in the random effects model, is calculated as the inverse of the variance + $TAU2$, divided by the sum of those. (Note that the original macro calculated $WRAND$ and then recalculated $WRAND$ as $WRAND$ divided by the sum of $WRAND$; for clarity, I have named the original calculation $WRAND_$.) The beta coefficients, multiplied by their weights in the random effects model, are summed across studies to calculate the pooled coefficients from the random effects model, $BETAR$. Its confidence bounds, $BLOWR$ and $BHIGHR$, are calculated using the relevant standard error, which is the square root of the inverse of the sum of $1/(VAR + TAU2)$. The pooled OR/RR for the random model, incorporating the $\&WT$ weight, are calculated along with its confidence bounds (ORR , $ORLOWR$, $ORHIGHR$ respectively). The Z-score, X^2 statistic, and p-value ($PCHI2R$) are then calculated to determine whether the pooled beta coefficient, from the random effects model, is significantly different from zero.

The remainder of the macro involves the calculations from the later Takkouche et al 2013 paper, which are beyond the scope of this paper, and a various reformatting of the calculations for presentation and output datasets.

Other than some minor changes (such as adding PUT statements to avoid numeric/character conversion notes), I tried to not alter the non-IML portions of the existing macro.

CONCLUSION

Hertzman and Spiegelman produced a very helpful, publicly available macro that implements the test for heterogeneity between studies when conducting a meta-analysis introduced in DerSimonian & Laird's 1986 paper. The goal of this paper was to create an adaptation of that macro that did not utilize Proc IML.

REFERENCES

DerSimonian, R. and N. Laird. 1986. "Meta-Analysis in Clinical Trials." *Controlled Clinical Trials*, 7.3:177-188.

Hertzmark, E. and D. Spiegelman. 2017. "The SAS METAANAL Macro." Available at <https://www.hsph.harvard.edu/donna-spiegelman/software/metaanal/>

Takkouche, B., Cadarso-Suarez, C., and D. Spiegelman. 1999. "Evaluation of Old and New Tests of Heterogeneity in Epidemiologic Meta-analysis." *American Journal of Epidemiology*, 150(2): 206-215.

Takkouche, B., Khudyakov P., Costa-Bouzas, J., and D. Spiegelman. 2013. "Confidence Intervals for Heterogeneity Measures in Meta-analysis." *American Journal of Epidemiology*, 178(6): 993-1004.

RECOMMENDED READING

While beyond the scope of this paper, readers may be interested in reviewing DerSimonian & Laird's new paper, "Meta-analysis in Clinical Trials Revisited", in the November 2015 issue of Contemporary Clinical Trials.

CONTACT INFORMATION

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

Venita DePuy
Owner and Principal Biostatistician, Bowden Analytics
Bowden.analytics@gmail.com

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

Other brand and product names are trademarks of their respective companies.

APPENDIX

```
/* =====
This macro is a modification of Hertzmark & Spiegelman's %METAANAL macro, available
at https://www.hsph.harvard.edu/donna-spiegelman/software/metaanal/

This modified macro adapts the original %METAANAL macro to use Base SAS instead of Proc IML
plus minor changes to prevent numeric/character conversion issues.
Modifications by Venita DePuy of Bowden Analytics, October 2017

This macro produces the Laird and DerSimonian weighted average of betas given s betas and their
variances.
includes changes made by Polyna Khudyakova (last change was done in August, 2015)
also, allows loglinear or linear models; user to give conf bounds instead of coeff and var/se
calculates measures of heterogeneity: R_I, CV_B and newly developed Rb (Aug. 2015)
===== */

%macro metaanal_noIML(
  coeff = T,          /* whether giving coeff or OR/RR. if OR/RR, no need for point est */
  beta = beta,        /* Input betas REQUIRED if COEFF=T */
  se_or_var = v,      /* whether you are giving the macro the standard error (s) or the
                        variances (v) of the coefficients if COEFF = T */
  varnamelcb =,       /* variable name for lower confidence bound of OR/RR if COEFF = F */
  varnameucb =,       /* variable name for upper confidence bound of OR/RR if COEFF = F */
  var = var,          /* need to give either beta & var or se OR varnamelcb & varnameucb */
  se = se,            /* variable name for variances */
  data =,             /* variable name for standard errors */
  studylab = studylab, /* Input data set REQUIRED */
  name =,             /* labels for each study REQUIRED */
  explabel=,          /* Name of variable of interest REQUIRED */
  outcomelabel=,      /* descriptive title of exposure REQUIRED */
  wt = 1,             /* descriptive title of outcome REQUIRED */
  outdat =,           /* Output data set */
  pooltype=random,    /* whether the underlying analysis is log-linear (logistic,
                        phreg, log-binomial, poisson) or not */
  notes=nonotes,
  printcoeff=F,
  loglinear=t,
  noprint =F,
  pvalueformat=pvalue6.4
);

/* upcasing true-false param */
%let coeff=%upcase(&coeff);
%let noprint=%upcase(&noprint);
%let printcoeff=%upcase(&printcoeff);
%let pooltype=%upcase(&pooltype);
%let loglinear=%upcase(&loglinear);
%let notes=%upcase(&notes);
%let se_or_var=%upcase(&se_or_var);

%local paramprob;
/* deal with potential problems */
%if %length(&data) eq 0
  or (%length(&beta) eq 0 and %length(&varnamelcb) eq 0 )
  /* or %length(&name) eq 0 */
  or %length(&studylab) eq 0
  or (&pooltype ne FIXED and &pooltype ne RANDOM and &pooltype ne BOTH)
  or %length(&explabel) eq 0 or %length(&outcomelabel) eq 0
  or (&se_or_var ne S and &se_or_var ne V)
%then %do;
  %let paramprob = 1;
  data _null_;
  %if %length(&data) eq 0 %then %do;
    put 'ERR'OR in macro run: No data set name was given for DATA';
  %end;
  %if %length(&beta) eq 0 and %length(&varnamelcb) eq 0 %then %do;
    put 'ERR'OR in macro run: No variable name was given for BETA or VARNAMELCB';
  %end;
  %if %length(&studylab) eq 0 %then %do;
```

```

        put 'ERR'OR in macro run: No value was given for STUDYLAB';
    %end;
    %if "&pooltype" ne "FIXED" and "&pooltype" ne "BOTH" and "&pooltype" ne "RANDOM" %then %do;
        put 'ERR'OR in macro call: Method for pooling (POOLTYPE) (fixed, random, or both)';
        put '    not specified.';
    %end;
    %if %length(&explabel) eq 0 %then %do;
        put 'ERR'OR in macro call: You did not describe the exposure in EXPLABEL';
    %end;
    %if %length(&outcomelabel) eq 0 %then %do;
        put 'ERR'OR in macro call: You did not describe the outcome in OUTCOMELABEL';
    %end;
    put 'The macro will stop.';

    file print;
    %if %length(&data) eq 0 %then %do;
        put 'ERR'OR in macro run: No data set name was given for DATA';
    %end;
    %if %length(&beta) eq 0 and %length(&varname) eq 0 %then %do;
        put 'ERR'OR in macro run: No variable name was given for BETA or VARNAME';
    %end;
    %if "&se_or_var" ne "S" and "&se_or_var" ne "V" %then %do;
        put 'ERR'OR in macro call: You did not specify whether you were';
        put '    giving the macro the variances or the standard errors';
        put '    of the coefficients.';
    %end;
    %if "&se_or_var" eq "V" and %length(&var) eq 0 %then %do;
        put 'ERR'OR in macro run: No variable name was given for VAR';
    %end;
    %if "&se_or_var" eq "S" and %length(&se) eq 0 %then %do;
        put 'ERR'OR in macro run: No variable name was given for SE';
    %end;
    %if %length(&studylab) eq 0 %then %do;
        put 'ERR'OR in macro run: No value was given for STUDYLAB';
    %end;
    %if "&pooltype" ne "FIXED" and "&pooltype" ne "BOTH" and "&pooltype" ne "RANDOM" %then %do;
        put 'ERR'OR in macro call: Method for pooling (POOLTYPE) (fixed, random, or both)';
        put '    not specified.';
    %end;
    %if %length(&explabel) eq 0 %then %do;
        put 'ERR'OR in macro call: You did not describe the exposure in EXPLABEL';
    %end;
    %if %length(&outcomelabel) eq 0 %then %do;
        put 'ERR'OR in macro call: You did not describe the outcome in OUTCOMELABEL';
    %end;
    put 'The macro will stop.';
%end;

%if %length(&wt) eq 0 %then %do; %let wt=1;
    data _null_;
    put "WARN" "ING in macro run: WT (the change of interest in &var was set to 1" ;
    run;
%end;

%if %length(&outdat) eq 0 %then %do;
    %if &noprnt ne F %then %do;
        data _null_;
        put 'WARN'ING in macro run: NOPRINT was not set to F, but no';
        put '    value was given for OUTDAT.';
        put 'The macro set NOPRINT to F.';
        file print;
        put 'WARN'ING in macro run: NOPRINT was not set to F, but no';
        put '    value was given for OUTDAT.';
        put 'The macro set NOPRINT to F.';
        run;
    %end;
%end;

%if &paramprob eq 1 %then %goto errend;

options &notes nosyntaxcheck formdlm='-';

```

```

data _datlist_ ; set &data end=_end_ ;
retain _nse _nvar _nbeta 0;
  %if "&loglinear" eq "T" %then %do;
    %if "&coeff" eq "T" %then %do;
      beta=&beta;
      %if "&se_or_var" eq "V" %then %do; var=&var; se=sqrt(&var); %end;
      %else %do; se=&se; var=&se * &se ; %end;
      odr=exp(&wt*&beta);
      lor=exp(&wt*(&beta-1.96*se));
      uor=exp(&wt*(&beta+1.96*se));
    %end;
    %if "&coeff" eq "F" %then %do;
      beta=mean(log(&varnameuch), log(&varnameuch));
      se=(log(&varnameuch)-beta)/1.96; var=se * se ;
      odr=exp(&wt * &beta);
      lor=&varnameuch;
      uor=&varnameuch;
    %end;
    label se='std error of study coefficient'
      odr='study OR/RR'
      lor='study lower 95% CL for OR/RR'
      uor='study upper 95% CL for OR/RR'
      beta='study coefficient' var='variance of study coefficient'
    ;
  %end;
  %else %do; /* loglinear=f */
    %if "&coeff" eq "T" %then %do;
      beta=&beta;
      %if "&se_or_var" eq "V" %then %do; var=&var; se=sqrt(var); %end;
      %else %do; se=&se; var=&se * &se ; %end;
    %end;
    %if "&coeff" eq "F" %then %do;
      beta=mean(&varnameuch, &varnameuch);
      se=(&varnameuch-beta)/1.96; var=se*se;
    %end;
    label se='std error of study coefficient'
      beta='study coefficient'
      var = 'variance of study coefficient'
    ;
  %end;
  if se ne . then _nse=_nse+1; if var ne . then _nvar=_nvar+1;
  if &beta ne . then _nbeta=_nbeta+1;
  if _end_ then do;
    *added PUT(xx, best.) statements to avoid numeric/character conversion n0te;
    call symput('_nse', trim(left(put(_nse,best))));
    call symput('_nvar', trim(left(put(_nvar,best))));
    call symput('_nbeta', trim(left(put(_nbeta,best))));
  end;
run;

%if &_nse eq 0 or &_nvar eq 0 or &_nbeta eq 0 %then %do;
  data _null_;
    put 'ERR'OR in macro run:;
    put ' There is not enough information to do the metaanalysis';
    if &_nbeta eq 0 then put " All values of the variable %upcase(&beta) are missing.";
    if &_nvar eq 0 then put " All values of the variable %upcase(&var) are missing.";
    if &_nse eq 0 then put " All values of the variable %upcase(&se) are missing.";
    file print;
    put ' There is not enough information to do the metaanalysis';
    if &_nbeta eq 0 then put " All values of the variable %upcase(&beta) are missing.";
    if &_nvar eq 0 then put " All values of the variable %upcase(&var) are missing.";
    if &_nse eq 0 then put " All values of the variable %upcase(&se) are missing.";
    call symput('paramprob', 1);
  run;
%end;
%if &paramprob eq 1 %then %goto errend;

```

```

data t0;
  retain sum_w;
  set _datlist_ nobs=dim;
dim=nrows(var);
df=dim-1;
call symput ('dim', strip(put(dim, best.)));
%do i = 1 %to &dim;
  if _N_ = &i then do;
    call symput("beta&i.", strip(put(beta, best.))); *IML: read all var{beta} into beta;
    call symput("label&i.", strip( &studylab)); *IML: read all var{&studylab} into labels;
    call symput("vari&i.", strip(put(var, best.))); *IML: read all var{var} into var;
  end;
%end;

odr = exp(&wt*beta);
se = sqrt(var);
ll = exp(&wt*beta-1.96*&wt*se);
uu = exp(&wt*beta+1.96*&wt*se);
lab = "str";

w = 1/var;
      *IML: w = (j(dim,1,1) /var);
sum_w = sum(sum_w, w);
varnum=_nbeta;
dummy=1;
run;

*merging sum of w=1/var onto all records;
data t1 (drop=varnum);
  retain betaf;
  merge t0 (drop=sum_w varnum)
        t0 (keep=dummy sum_w varnum where=(varnum=&dim)) ;
  by dummy;
  vsrf = 1/sum_w;
  wfixed = w/sum_w;
      *IML: vsrf = 1/sum(w);
      *IML: wfixed=w/sum(w);

  betaf_i = beta * wfixed;
  betaf = sum(betaf_i, betaf);
  if last.dummy then lastflag=1;
  run;
      *IML: betaf = (wfixed`*beta) ;

*merging betaf values onto all records;
data t2 (drop=lastflag q_i sqsumw_i sumw_i);
  retain q sqsumw sumw;
  merge t1 (where=(lastflag=1) keep = betaf dummy lastflag)
        t1 (drop=betaf );
  by dummy;

sef = sqrt(vsrfr);
blowf = betaf - 1.96*sef;
bhighf = betaf + 1.96*sef;

orf = exp(&wt*betaf);
orlowf = exp(&wt*blowf);
orhighf = exp(&wt*bhighf);

zscoref = betaf / sef;
chi2f = zscoref*zscoref;
pchi2f = 1 - probchi(chi2f,1);
      *IML: wm = diag(w);

q_i = (beta - betaf)**2 * w;
q = sum(q_i, q);
      *IML: q = (( beta - betaf )`)* wm * (beta - betaf );

sqsumw_i = w*w;
sumw_i = w;
sqsumw = sum(sqsumw_i, sqsumw);
sumw = sum(sumw_i, sumw);
if last.dummy then lastflag2=1;
run;
      *IML: sqsumw=w`*w;
      *IML: sumw=sum(w);

```

```

*adding q, sqsumw to all records;
data t3 (drop=lastflag2);
    retain sumwrand;
    merge t2 (where=(lastflag2=1) keep = q sqsumw sumw lastflag2 dummy )
           t2 (drop= q sqsumw sumw );
    by dummy;

    tau2 = max(0, (q-df)/(sumw - (sqsumw/sumw)) );
    *IML: tau2 = max(0, (q-(dim-1))/(sum(w) - (w`*w/sum(w)) ));
    wrand_ = 1/(var + tau2);
    *IML: wrand = (j(dim,1,1) / ( var + tau2 ));
    phet = 1-probchi(q,df);

    sumwrand = sum(wrand_, sumwrand);
    *renamed from wrand for clarity as wrand is divided by sum(wrand) later;
    if last.dummy then lastflag=1;
    run;

*adding sum (wrand_) to all records;
data t4 (drop=lastflag );
    retain betar;
    merge t3 (where=(lastflag =1) keep = sumwrand lastflag dummy )
           t3 (drop= sumwrand );
    by dummy;
    vsrr=1/sumwrand;
    wrand = wrand_/sumwrand;
    *IML: vsrr = 1/sum(wrand);
    *IML: wrand = wrand/sumwrand;

    betar_i = beta * wrand;
    betar = sum(betar_i, betar);
    *IML: betar= (wrand`*beta) ;
    if last.dummy then lastflag2=1;
    run;

*merging betar values onto all records;
data t5(drop=lastflag2 dgq_b_i rb_);
    retain rb_ dgq_b 0;
    merge t4 (where=(lastflag2=1) keep = betar dummy lastflag2)
           t4 (drop= betar );
    by dummy;

    ser = sqrt(vsrr);
    blowr = betar - 1.96*ser;
    bhighr = betar + 1.96*ser;

    orr= exp(&wt*betar);
    orlowr = exp(&wt*blowr);
    orhighr = exp(&wt*bhighr);

    zscorer = betar / ser;
    chi2r = zscorer*zscorer;
    pchi2r = 1 - probchi(chi2r,1);

    *based on modification to original macro by PK in August 2015;

    ab = var*(sumw - (sqsumw/sumw)) ;
    www = (sumw - (sqsumw/sumw)) ;
    rb_ = rb_ + tau2/(tau2 + var);
    bb = 1/ (q + ab - df)**2;
    rb = rb_/ &dim;
    *IML: ab =var*(sum(w) - (w`*w/sum(w)) );
    *IML: www = (sum(w) - (w`*w/sum(w)) );
    *IML: do i=1 to dim, rb=rb+tau2/(tau2+var[i]);
    *IML: bb[i,1]=1/((q+qb[i,1]-df**2), end;

    dgq_b_i = ab*bb/&dim;
    dgq_b = sum(dgq_b_i, dgq_b_i);
    if last.dummy then lastflag=1;
    run;
    *IML: dgq_b=ab`*bb/dim;;

*adding dgq_b and rb to all records;
data t6 (drop=lastflag);
    merge t5 (where=(lastflag =1) keep = rb dgq_b lastflag dummy )
           t5 (drop= rb dgq_b );
    by dummy;
    dim = &dim;
    run;

```

```

data _out_;
    set t6 (keep = betaf betar sef ser blowf bhighf blowr bhighr zscoref zscorer chi2f chi2r
pchi2f pchi2r
            sqsumw sumw w tau2 orf orr orlowf orlowr orhighf orhighr q df phet
            rb dgq b /*fwt rwt*/ vsrf vsrr dim );
            *list from original macro, but these 2 vars not yet created;
run;

data _labs_;
    length labels $100;
    %do i = 1 %to &dim;
        labels = "&&label&i.";
        output;
    %end;
run;

proc datasets ;
    delete t0 t1 t2 t3 t4 t5 t6;
run; quit;run;

%errend:
    options notes syntaxcheck;

*remainder of macro is taken directly from Hertzmark & Speigelman_s macro, including Polyna
Khudyakov revisions;

%if &notes eq NOTES %then %do;
    proc print data=_labs_ title2 '_labs_'; run;
    proc print data=_out_ (obs=5) title2 'out'; run;
%end;

data _out1_;
    set _out_;
    if _n_ eq 1;
        *added PUT(xx, best.) statements to avoid numeric/character conversion n0te;
    call symput ('_tau2_', trim(left(put(tau2,best.))));
run;

data _labs_;
    set _labs_;
    length _name_ $20;
    _name_ = "study_"||left(put(_n_, z2.0));
run;
proc transpose data=_labs_ out=_labs_;
    var labels;
run;

%if &notes eq NOTES %then %do;
    proc print data=_labs_ title2 '_labs_'; run;
%end;

data _summary_;
    * length name $35;
    merge _out1_ _labs_;
    drop _name_;
    if tau2 gt 0 or vsrf gt 0 then do;
        r_i=tau2 / (tau2 + dim*vsrf); pr_i=round(100*r_i, .1);
        end;
    if betar ne 0 then do; cv_betaw=sqrt(tau2) / abs(betar);
    pcv_betaw=round(100*cv_betaw, .1);
    end;

    call symput('r_i', trim(left( put( r_i , best.) )));
    call symput('pr_i', trim(left( put( pr_i , best.) )));
    call symput('cvbetaw', trim(left( put( cv_betaw , best.) )));
    call symput('pcvbetaw', trim(left( put( pcv_betaw , best.) )));
    call symput('df', trim(left( put( df , best.) )));

```

```

label
  betaf='pooled coeff from fixed effects model'
  betar='pooled coeff from random effects model'
  sef='pooled SE from fixed effects model'
  ser='pooled SE from random effects model'
  blowf='pooled lower 95% CL for coeff from fixed effects model'
  blowr='pooled lower 95% CL for coeff from random effects model'
  bhighf='pooled upper 95% CL for coeff from fixed effects model'
  bhighr='pooled upper 95% CL for coeff from random effects model'
  zscoref='pooled z-score from fixed effects model'
  zscorer='pooled z-score from random effects model'
  chi2f='pooled Chi-squared from fixed effects model'
  chi2r='pooled Chi-squared from random effects model'
  pchi2f='p-value based on pooled Chi-squared from fixed effects model'
  pchi2r='p-value based on pooled Chi-squared from random effects model'
  orf='pooled OR from fixed effects model'
  orr='pooled OR from random effects model'
  orlowf='pooled lower 95% CL for OR from fixed effects model'
  orlowr='pooled lower 95% CL for OR from random effects model'
  orhighf='pooled upper 95% CL for OR from fixed effects model'
  orhighr='pooled upper 95% CL for OR from random effects model'
  q='Q-statistic'
  df='degrees of freedom for Q-statistic'
  phet='p-value for test of heterogeneity'
  r_i='between-studies fraction of variance'
  pr_i='between-studies percent of variance'
  cv_betaw='between-studies coeff of variance, as fraction'
  pcv_betaw='between-studies coeff of variance, as percent'
  tau2='estimate of between-studies variance'
;
call symput('betar', trim(left( put( betar , best.) )));
call symput('sqsumw', trim(left( put( sqsumw , best.) )));
call symput('sumw', trim(left( put( sumw , best.) )));
call symput('qstat', trim(left( put( q , best.) )));
run;

%if &notes eq NOTES %then %do;
  proc print data=_summary_ (obs=5); title2 '_summary_'; run;
%end;

data _datlist_; set _datlist_;
  if var gt 0 then fwtx=1/var;
  if var gt 0 or &_tau2_ gt 0 then rwtx=1/(var+ &_tau2_);
  if var gt 0 then do; w1=1/var; w2=w1/var; w3=w2/var; end;
run;

%if &notes eq NOTES %then %do;
  proc print; title2 '_datlist_ first time';
%end;

proc means noprint data=_datlist_; var fwtx rwtx w1 w2 w3 ;
  output out=_sums_ sum=sumf sumr w1 w2 w3 n(fwtx)=numstud ;
run;

data _datlist_;
  if _n_ eq 1 then set _sums_; set _datlist_;
  fwt=fwtx/sumf; rwt=rwtx/sumr;
  label fwt='weight in fixed effects model'
        rwt='weight in random effects model'
  ;
run;

data _summary_;
  if _n_ eq 1 then set _sums_; set _summary_;
  tau2=&_tau2_;
  ;
  betahat=betaf; betahat2=betahat*betahat;
  betahatr=betar; betahatr2=betahatr*betahatr;
  varq=2*(numstud-1) +4*(w1-w2/w1)*tau2 + 2*(w2-2*w3/w1 +w2**2/w1**2)*tau2*tau2;
  seq=sqrt(varq);
  if &qstat ne 0 then varlogq=varq/&qstat/&qstat;

```

```

selogq=sqrt(varlogq);
lclq=exp(log(&qstat)-1.96*selogq); uclq=exp(log(&qstat)+1.96*selogq);
qlow=max(0,&qstat-1.96*seq); qhi=&qstat+1.96*seq;
c1=numstud-1; c2=numstud*&sqsumw/(&sumw*&sumw)-1;
rilow=1-(c1-c2)/(&qstat-1.96*seq -c2); rihi=1-(c1-c2)/(&qstat+1.96*seq -c2);
varri=(c1-c2)**2 / (&qstat-c2)**4 * varq;
lcri=r_i-1.96*sqrt(varri); ucrlri=r_i+1.96*sqrt(varri);
lclri=max(0,100*lcri); uclri=min(100,100*ucrlri);
rilow=max(0,100*rilow); rihi=min(100,100*rihi);
vartau2=varq/(w1-w2/w1)**2;
if tau2 gt 0 then vartau=vartau2/4/tau2;
if tau2 gt 0 then varcvb=vartau2/(4*tau2*betahatr2)+vsrr*tau2/(betahatr2*betahatr2);
/*varlogcvb=varcvb/cv_betaw/cv_betaw;
selogcvb=sqrt(varlogcvb);
cvblow=exp(log(cv_betaw)-1.96*selogcvb); cvbhi=exp(log(cv_betaw)+1.96*selogcvb);*/
cvblow=max(0,cv_betaw-1.96*sqrt(varcvb)); cvbhi=cv_betaw+1.96*sqrt(varcvb);
* Aug. 2015 (PK);
varrb=(dgq_b**2)*varq;
lcrb=rb-1.96*sqrt(varrb); ucrrb=rb+1.96*sqrt(varrb);
lclrb=max(0,100*lcrb); uclrb=min(100, 100*ucrrb);
lrb=round(100*rb, .1);
if _n_ eq 1 then do;
    call symput('qlow', trim(left( put( qlow , best.) )));
    call symput('qhi', trim(left( put( qhi , best.) )));
    call symput('rilow', trim(left( put( rilow , best.) )));
    call symput('rihi', trim(left( put( rihi , best.) )));
    call symput('cvblow', trim(left( put( cvblow , best.) )));
    call symput('cvbhi', trim(left( put( cvbhi , best.) )));
    call symput('lclq', trim(left( put( lclq , best.) )));
    call symput('uclq', trim(left( put( uclq , best.) )));
    call symput('lclri', trim(left( put( lclri , best.) )));
    call symput('uclri', trim(left( put( uclri , best.) )));
    call symput('lclrb', trim(left( put( lclrb , best.) )));
    call symput('uclrb', trim(left( put( uclrb , best.) )));
    call symput('lrb', trim(left( put( lrb , best.) )));
end;
run;

%if &notes eq NOTES %then %do;
    proc print data=_datlist_ (obs=5); title2 'datlist'; run;
    title2;
%end;

data _null_; set _out_;
q=round(q, .0001);
format phet &pvalueformat;
length phetc $7.; phetc=put(phet, &pvalueformat);
call symput('phet', trim(left( phetc )));
tau2=round(tau2, .0001);
call symput('tau2', trim(left( put( tau2 , best.) )));
run;

title4 "Studies of &explabel and &outcomelabel";
title5 'listing of input data';
proc print label data=_datlist_;
    %if "&loglinear" eq T %then %do;
        var &studylab &beta se &var odr lor uor fwt rwt;
        format odr lor uor fwt rwt 8.2;
    %end;
    %else %do;
        var &studylab &beta se fwt rwt;
    %end;
run;

data _niceout_; set _summary_;
length shortname $10 varlabel $70 varvalue $60 varname $10 vv1 vv3 vv5 $8 vv2 vv4 $2 vv6 $1 ;
pvalue=.; format pvalue &pvalueformat;
*modtype='f';
%if "&loglinear" eq "T" %then %do;
    varname='orf'; varlabel='OR/RR from fixed effects model (95% CI)';
    shortname='OR/RR (F)';

```

```

    vv1=put(orf, 8.2); vv2=' ('; vv3=put(orlowf, 8.2); vv4=', '; vv5=put(orhighf, 8.2);
    vv6=')';
    varvalue=compress(vv1) || vv2 || compress(vv3) || vv4 || compress(vv5) || vv6 ;
    %end;
%else %do;
    varname='coefff'; varlabel='Coefficient from fixed effects model (SE)';
    shortname='Coeff';
    vv1=put(betaf, 8.4); vv2=' ('; vv3=put(sef, 8.4); vv4=', '; vv6=')';
    varvalue=compress(vv1) || ' ' || compress(vv2 || vv3 || vv6);
    %end;
pvalue=pchi2f;
output;
%if "&printcoeff" eq "T" %then %do;
    varname='betaf'; varlabel='Coefficient from fixed effects model';
    shortname='Coeff. (F)';
    vv1=put(betaf, 8.2); vv3=put(blowf, 8.2); vv5=put(bhighf, 8.2); vv4=', '; vv6=')';
    varvalue=compress(vv1) || vv2 || compress(vv3) || vv4 || compress(vv5) || vv6 ;
    pvalue=pchi2f;
    output;
%end;
*modtype='r';
%if "&loglinear" eq "T" %then %do;
    varname='orr'; varlabel='OR/RR from random effects model (95% CI)';
    shortname='OR/RR (R)';
    vv1=put(orr, 8.2); vv2=' ('; vv3=put(orlowr, 8.2); vv4=', '; vv5=put(orhighr, 8.2);
    vv6=')';
    varvalue=compress(vv1) || vv2 || compress(vv3) || vv4 || compress(vv5) || vv6 ;
    %end;
%else %do;
    varname='coeff'; varlabel='Coefficient from random effects model (SE)';
    shortname='Coeff';
    vv1=put(betar, 8.4); vv2=' ('; vv3=put(ser, 8.4); vv4=', '; vv6=')';
    varvalue=compress(vv1) || ' ' || compress(vv2 || vv3 || vv6);
    %end;
pvalue=pchi2r;
output;
%if "&printcoeff" eq "T" %then %do;
    varname='betar'; varlabel='Coefficient from random effects model';
    shortname='Coeff. (R)';
    vv1=put(betar, 8.2); vv3=put(blowr, 8.2); vv5=put(bhighr, 8.2); vv4=', '; vv6=')';
    varvalue=compress(vv1) || vv2 || compress(vv3) || vv4 || compress(vv5) || vv6 ;
    pvalue=pchi2r;
    output;
%end;
/* statistics relating to heterogeneity */
varname="qstat"; varlabel="Q-statistic for heterogeneity with &df degree(s) of freedom";
shortname='Q';
vv1=put(&qstat, 10.4);
* The change below done by P. Khudyakov (July 21st, 2015);
*****;
* Since it is assumed that Q statistics has Chi square distribution,
which is skewed and by applying the Log-transformation, we can get approximately normal
distribution and then to calculate CI as for normal distribution.
When the value of Q is too small, the log-transformation does not work.
Therefore, to avoid further confusion, we decided to exclude CIs of Q from the output;
*****;
/* vv3=put(&qlow, 10.4); vv5=put(&qhi, 10.4);
varvalue=compress(vv1)|| vv2 || compress(vv3) || vv4 || compress(vv5) || vv6 ;
*/
varvalue=compress(vv1) ;
pvalue=phet;
output;
varname="tau2"; varlabel='Estimate of between-studies variance (tau squared)';
shortname='tau2';
varvalue=left(put(tau2, 10.4));
pvalue=. ; *pvalue='.';
output;
if tau2 gt 1e-10 then do;
    varname='ri'; varlabel='Between-studies variance as a percent of total variance (r(i))';
    shortname='r(i)';
    vv1=put(&pr_i, 5.1); vv3=put(&lclri, 5.1); vv5=put(&uclri, 5.1); vv4=', '; vv6=')';

```

```

varvalue=compress(vv1) || vv2 || compress(vv3) || vv4 || compress(vv5) || vv6 ;
pvalue=.;
output;
varname='cvb'; varlabel='SQRT(Between-studies variance)/coeff. from random-effects model
(CVB)';
shortname='CVB';
vv1=put(&cvbetaw, 7.3); vv3=put(&cvblow, 7.3); vv5=put(&cvbhi, 7.3); vv4=', ';
vv6=');
varvalue=compress(vv1) || vv2 || compress(vv3) || vv4 || compress(vv5) || vv6 ;
pvalue=.;
output;
* Aug. 2015 (PK);
varname='rb'; varlabel='New between-studies variance as a percent of total variance
(r(b))';
shortname='r(b)';
vv1=put(lrb, 7.1); vv3=put(&lclrb, 7.1); vv5=put(&uclrb, 7.1); vv4=', '; vv6=');
varvalue=compress(vv1) || vv2 || compress(vv3) || vv4 || compress(vv5) || vv6 ;
pvalue=.;
output;
end;

*modtype='f';
drop vv1-vv6;
run;

%if &notes eq NOTES %then %do;
proc print data=_datlist_ (obs=5); run;
%end;

data _niceout_ ; set _niceout_ ;
label varlabel='Description'
shortname='Statistic'
varvalue='Value (95% CI)'
pvalue='P'
testing='Hypothesis being tested'
pvaluec='P (testing hypothesis)'
;
length pvaluec $20 testing $75 ;
/*
if varlabel eq: 'OR/RR' then do;
pvaluec=put(pvalue, &pvalueformat) || '(Is OR/RR different from 1?)';
testing='Is OR/RR different from 1?';
end;
*/
if shortname eq: 'OR/RR (F)' then do;
pvaluec=put(pvalue, &pvalueformat) || '(Is OR/RR different from 1?)';
testing='Is OR/RR different from 1? (Fixed effects model)';
end;
else if shortname eq: 'OR/RR (R)' then do;
pvaluec=put(pvalue, &pvalueformat) || '(Is OR/RR different from 1?)';
testing='Is OR/RR different from 1? (Random effects model)';
end;
else if varname eq 'betaf' then testing='Is coeff. different from 0? (Fixed effects model)';
else if varname eq 'betar' then testing='Is coeff. different from 0? (Random effects model)';
else if varlabel eq: 'Q' then do;
pvaluec=put(pvalue, &pvalueformat) || '(Is there heterogeneity among the studies?)';
testing='Is there heterogeneity among the studies?';
end;
else do; pvaluec=' '; testing=' '; end;
run;

%if &notes eq NOTES %then %do;
proc print data=_niceout_ (obs=5); title2 '_niceout_'; run;
title2 ;
%end;

title4 "Studies of &explabel and &outcomelabel";

```

```

proc print data=_niceout_label ; var shortname varvalue pvalue testing;
  %if "&pooltype" eq "RANDOM" %then %do;
    where index(varlabel, 'fixed') le 0;
  %end;
  %if "&pooltype" eq "FIXED" %then %do;
    where index(varlabel, 'random') le 0;
  %end;
proc print data=_niceout_label ; var shortname varlabel;
  %if "&pooltype" eq "RANDOM" %then %do;
    where index(varlabel, 'fixed') le 0;
  %end;
  %if "&pooltype" eq "FIXED" %then %do;
    where index(varlabel, 'random') le 0;
  %end;

  %if %length(&outdat) ne 0 %then %do;
    data &outdat._d; set _datlist_
    %if "&loglinear" eq T %then %do;
      (keep= &studylab &beta se &var odr lor uor fwt rwt)
    %end;
    %else %do;
      (keep= &studylab &beta se fwt rwt)
    %end;
  ;
run;

data &outdat._p;
  set _niceout_ (keep=shortname varvalue pvalue testing varlabel) ; run;
%end;

proc datasets mt=data nolist;
  delete _niceout_ _summary_ _datlist_ _sums_ _out_ _labs_ ; run; quit;run;

%mend;

```