The Stata Journal

The *Stata Journal* publishes reviewed papers together with shorter notes or comments, regular columns, book reviews, and other material of interest to Stata users. Examples of the types of papers include 1) expository papers that link the use of Stata commands or programs to associated principles, such as those that will serve as tutorials for users first encountering a new field of statistics or a major new technique; 2) papers that go "beyond the Stata manual" in explaining key features or uses of Stata that are of interest to intermediate or advanced users of Stata; 3) papers that discuss new commands or Stata programs of interest either to a wide spectrum of users (e.g., in data management or graphics) or to some large segment of Stata users (e.g., in survey statistics, survival analysis, panel analysis, or limited dependent variable modeling); 4) papers analyzing the statistical properties of new or existing estimators and tests in Stata; 5) papers that could be of interest or usefulness to researchers, especially in fields that are of practical importance but are not often included in texts or other journals, such as the use of Stata in managing datasets, especially large datasets, with advice from hard-won experience; and 6) papers of interest to those who teach, including Stata with topics such as extended examples of techniques and interpretation of results, simulations of statistical concepts, and overviews of subject areas.

The *Stata Journal* is indexed and abstracted by *CompuMath Citation Index*, *Current Contents/Social and Behavioral Sciences*, *RePEc: Research Papers in Economics*, *Science Citation Index Expanded* (also known as *SciSearch*, *Scopus*, and *Social Sciences Citation Index*.

For more information on the *Stata Journal*, including information for authors, see the webpage

http://www.stata-journal.com

Copyright © 2013 by StataCorp LP

Simulation-based sample-size calculation for designing new clinical trials and diagnostic test accuracy studies to update an existing meta-analysis

Michael J. Crowther
Department of Health Sciences
University of Leicester
Leicester, UK
michael.crowther@le.ac.uk

Sally R. Hinchliffe
Department of Health Sciences
University of Leicester
Leicester, UK

Alison Donald
Department of Health Sciences
University of Leicester
Leicester, UK

Alex J. Sutton
Department of Health Sciences
University of Leicester
Leicester, UK

Abstract. In this article, we describe a suite of commands that enable the user to estimate the probability that the conclusions of a meta-analysis will change with the inclusion of a new study, as described previously by Sutton et al. (2007, *Statistics in Medicine* 26: 2479–2500). Using the `metasim` command, we take a simulation approach to estimating the effects in future studies. The method assumes that the effect sizes of future studies are consistent with those observed previously, as represented by the current meta-analysis. Two-arm randomized controlled trials and studies of diagnostic test accuracy are considered for a variety of outcome measures. Calculations are possible under both fixed- and random-effects assumptions, and several approaches to inference, including statistical significance and limits of clinical significance, are possible. Calculations for specific sample sizes can be conducted (by using `metapow`). Plots, akin to traditional power curves, can be produced (by using `metapowplot`) to indicate the probability that a new study will change inferences for a range of sample sizes. Finally, plots of the simulation results are overlaid on extended funnel plots by using `extfunnel`, described in Crowther, Langan, and Sutton (2012, *Stata Journal* 12: 605–622), which can help to intuitively explain the results of such calculations of sample size. We hope the command will be useful to trialists who want to assess the potential impact new trials will have on the overall evidence base and to meta-analysts who want to assess the robustness of the current meta-analysis to the inclusion of future data.

Keywords: st0304, metasim, metapow, metapowplot, meta-analysis, diagnostic test, sample size, evidence-based medicine

1   Introduction

Sutton et al. (2007) argued that following the completion of a new randomized trial, the updated meta-analysis containing the new study would potentially be of more interest where multiple studies of the same topic exist in certain contexts. However, this goes against findings that many trialists do not consider previous trials, formally or informally, when designing new trials (Cooper, Jones, and Sutton 2005).

Relatively recently, formal methodology to assess the ability of new trials to affect the conclusions of an updated meta-analysis were developed (Sutton et al. 2009b). These methodologies were piloted in a study that applied them retrospectively to several clinical contexts (Goudie et al. 2010). A coherent framework for designing, analyzing, and reporting evidence that used such methods has also been described in Sutton, Cooper, and Jones (2009a). Very recently, the methods have been adapted for diagnostic test accuracy by Hinchliffe et al. (2013) and for cluster-randomized controlled trials by Rotondi and Donner (2012). Both Bayesian (Sutton et al. 2007) and frequentist (Goudie et al. 2010) implementations of the general approach have been considered.

Here we describe a collection of three commands to implement the frequentist version of the methodology in the contexts of (two-arm) randomized controlled trials and diagnostic test accuracy. `metasim` simulates data for future studies of a specified sample size by using predictions based on a meta-analysis of the existing evidence. Although this command can be used on its own, as described in sections 2.2 and 3, it is primarily designed to be used as a subroutine that is called by `metapow` (sections 2.2 and 4). `metapow` calculates the probability that a future study with a sample size specified by the user will change the inferences of an existing meta-analysis.[1] Several alternative approaches to inference can be specified, including both statistical significance and limits of clinical significance. `metapowplot` (sections 2.2 and 5) presents a graph of power for a range of sample sizes for the new study by repeatedly invoking `metapow`.

The structure of the remainder of this article is as follows: Section 2 describes the methods implemented in the three commands. Sections 3, 4, and 5 describe the syntax for `metasim`, `metapow`, and `metapowplot`, respectively. In section 6, plots of the simulated study results are overlaid on the previously described command `extfunnel` (Crowther, Langan, and Sutton 2012). This can help to intuitively explain the results of the power calculations through the use of boundary contours where inferences of the meta-analysis will change, indicating the effect size and precision combinations of future studies that would change inferences of the meta-analysis (Langan et al. 2012). Section 7, the discussion, concludes the article.

---

1. We refer to the probability that inferences of the meta-analysis will change as "power" throughout the remainder of the article, although we acknowledge this is not what is usually referred to as "power" in a single-study context.

2   Methods

2.1   Overview of methods

Simulation methods to establish appropriate sample sizes are often used as an alternative to closed-form solutions when complex analyses need to be carried out (Feiveson 2002) (for example, for analyses that include models with random effects). The approach suggested by Sutton et al. (2007) focuses not on the simulated study itself but on the modified meta-analysis, including the simulated study, because in some contexts, the results of the updated meta-analysis will be of more interest than those of the study on its own. Below is the nontechnical summary of the process as described by Sutton et al. (2007).

1. From a meta-analysis of the existing studies, a distribution for the chosen outcome measures in a new clinical trial or diagnostic test accuracy study is derived. An estimate for the outcome measure from this distribution is then sampled, representing the underlying effect in the new (simulated) study.

2. Data representing the new study are generated stochastically according to the estimate sampled in step 1 for a specified sample size.

3. These simulated study data are then added to the existing meta-analysis, which is then re-meta-analyzed.

4. The hypothesis test, on which decisions are to be based, is then considered. Whether the null is retained or rejected in favor of the alternative hypothesis at a specified level of statistical significance is recorded.

5. Steps 2–4 are repeated a large number of times (N), and the outcome of the hypothesis test is noted each time.

6. Power is estimated by calculating the proportion of the N simulations in which the null hypothesis is rejected.

7. The procedure is iterative: the sample size for a new study specified in step 2 is modified and steps 2–6 repeated until the desired level of power is achieved.

2.2   Overview of software

Figure 1 presents a schematic representation of the relationship between the Stata commands described in this article and previously described commands. It has already been explained that `metapowplot` calls `metapow`, which in turn calls `metasim`. Additionally, however, because others have written excellent routines for conducting meta-analysis in Stata, the preexisting `metan` command (Harris et al. 2008) is called by `metapow` to conduct meta-analyses of two-arm comparative study data, such as data from randomized controlled trials. Similarly, `metandi` (Harbord and Whiting 2009) and `midas` (Dwamena

2007) are both called by `metapow` for conducting meta-analyses of diagnostic test accuracy studies. Two commands are used because convergence issues with the bivariate diagnostic model are well documented (Rabe-Hesketh, Skrondal, and Pickles 2005). In addition, because both routines use different estimation algorithms, `midas` is invoked if `metandi` fails to converge. The command `extfunnel`, described in Crowther, Langan, and Sutton (2012), can be used to further illustrate the simulation results, and we show in section 6 how output from `metapow` can be overlaid on the plots produced by `extfunnel`.
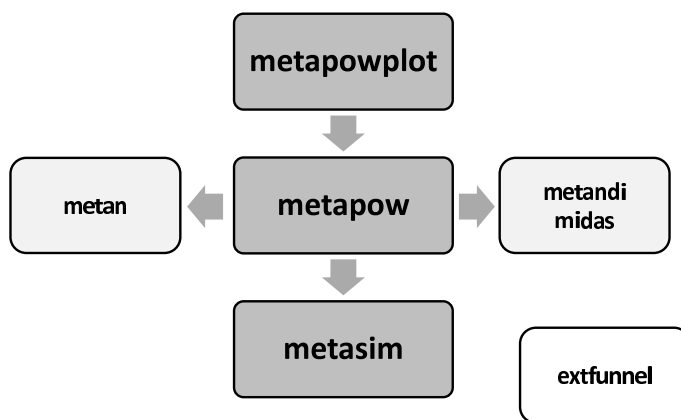
Figure 1. Software relationship diagram: arrows denote the calling of a command

metasim

`metasim` simulates a specified number of new studies based on the estimate obtained from a preexisting meta-analysis, assuming the effect size seen in the new study will be consistent with the existing studies in the meta-analysis. The command can be used independently, but it was designed to be used in conjunction with `metapow` (see section 4).

`metasim` will simulate data for a new study represented by the values for each of the variables entered in the variable list. These are saved in a Stata data file, `temppow.dta`, in the specified working directory.

metapow

`metapow` implements an approach to estimating the power of a study based on the evidence-based approach to sample-size determination for adding new studies to a meta-analysis of two-arm randomized controlled trials and diagnostic accuracy studies described in Sutton et al. (2007) and Hinchliffe et al. (2013), respectively. Power is determined through simulation, with data for new studies being generated with `metasim`.

As well as estimating the power of the updated meta-analysis including the new study, `metapow` can also estimate the power of the new study when analyzed on its own. The results of individual simulations are stored in a file, `temppow2.dta`, located in the specified working directory. While this function can be used directly to estimate the power for particular sample sizes, the higher-level command `metapowplot` uses this command to construct power curves across different sample sizes.

metapowplot

`metapowplot` produces a plot of the power values for a range of sample sizes. The command calls on `metapow`, which calculates power for a single sample size. `metapow` in turn calls on `metasim`, which simulates new studies by using the results of the existing meta-analysis.

Users need to input a minimum and maximum sample size for which they want to calculate a power estimate. The power estimates are stored with their confidence intervals (CIs) in a file called `temppow3.dta` within the working directory.

3   The metasim command

3.1   Syntax

metasim *varlist*, n(*integer*) es(*numlist*) var(*numlist*)

    <u>type</u>(clinical|diagnostic) [<u>measure</u>(or|rr|rd|nostandard|dor|ss)

    p(*real*) r(*real*) <u>studies</u>(*integer*)

    <u>mod</u>el(fixed|fixedi|random|randomi|bivariate) <u>tausq</u>(*numlist*)

    dist(normal|t) corr(*real*) path(*string*)]

The dataset should contain the data for the existing studies with variable names that are consistent with those entered in *varlist*. The user should input a maximum of six variables. For trials with a binary outcome, four variables are required: these correspond to the number of events and nonevents in the experimental group followed by those of the control group. And for continuous outcomes, six variables should be entered: sample size, mean, and standard deviation of the experimental group followed by those of the control group. For diagnostic studies, four variables are required: the true positives, false positives, false negatives, and true negatives.

3.2   Options

n(*integer*) relates to the number of patients in the new study. If users simulate a new clinical trial, then `n()` specifies the number of patients in the control group. If users simulate a new diagnostic accuracy study with sensitivity and specificity as the outcome measure of accuracy, then `n()` is the number of diseased patients. If users simulate a new diagnostic accuracy study with the diagnostic odds ratio (DOR), then `n()` is the number of positive test results. `n()` is required.

es(*numlist*) specifies the pooled estimates from the meta-analysis of existing studies. If using the odds ratio (OR), the DOR, or relative risk (RR), then users need to specify ln(OR), ln(DOR), or ln(RR) estimates, respectively. If using sensitivity and specificity, then users need to specify logit(sensitivity) and logit(specificity), in that order. `es()` is required.

var(*numlist*) specifies the variances for `es()`. Two values should be entered when using sensitivity and specificity. `var()` is required.

type(clinical | diagnostic) specifies the type of new study that the user would like to simulate: a two-arm clinical trial or a diagnostic test accuracy study. `type()` is required.

measure(or | rr | rd | nostandard | dor | ss) specifies the outcome measure used in the meta-analysis to pool the results. The OR (`or`), RR (`rr`), risk difference (`rd`), and unstandardized mean difference (`nostandard`) can only be used when simulating a new clinical study. The DOR (`dor`) and sensitivity and specificity (`ss`) can only be used when simulating a new diagnostic accuracy study. The default for a `type(clinical)` study with four variables entered into the *varlist* is `rr`; the default for a `type(clinical)` study with six variables entered into the *varlist* is `nostandard`; and the default for a `type(diagnostic)` study is `ss`.

p(*real*) is the estimated event rate in the control group in a simulated, new clinical study. When users simulate a new diagnostic accuracy study, this is the estimated probability of being diseased given a positive result in the new study. When this option is not specified, `metasim` will calculate this value by averaging the probabilities across the studies included in the dataset in memory. Note that `p()` is only relevant in the diagnostic framework when `dor` is used as the option in `measure()`.

r(*real*) is the ratio of patients in the control group to the treatment group in a simulated, new clinical study. When users simulate a new diagnostic accuracy study, this is the ratio of diseased to healthy people if using sensitivity and specificity and is the ratio of positive to negative results if using the DOR. The default is r(1).

studies(*integer*) specifies the number of new studies to be simulated and included in the updated meta-analysis. The default is studies(1). When more than one study is specified, each is assumed to have the same sample size.

model(fixed | fixedi | random | randomi | bivariate) defines the type of model used to meta-analyze the preexisting data. The default is model(fixed) unless the outcome measure is the nonstandardized mean difference, in which case the default is model(fixedi). The model(fixedi) option specifies a fixed-effects model by using the inverse-variance method. The model(random) option uses the random-effects DerSimonian and Laird method, taking the estimate for heterogeneity from the Mantel–Haenszel method. The model(randomi) option specifies a random-effects model by using the method of DerSimonian and Laird, with the estimate of heterogeneity being taken from the inverse-variance fixed-effects model. All the above options call on the metan command within metasim. The final option is the bivariate random-effects model (model(bivariate)). This method calls on a combination of the metandi and midas commands (a variable is created to indicate which has been used for each simulation). It may only be specified when simulating a new diagnostic accuracy study.

tausq(*numlist*) is the measure of between-study variance taken from the preexisting meta-analysis. The default is tausq(0). If measure(ss) is specified, then two values must be entered for tausq(). If a random-effects model is selected and the value for tausq() is still 0, then a warning message will appear to notify the user, but the command will continue to run.

dist(normal | t) specifies the distribution of effect sizes used to sample a value to simulate a new study. The default for model(random) and model(randomi) is a predictive distribution based on the *t* distribution (dist(t)), allowing for heterogeneity between studies (and the uncertainty in the heterogeneity). The default for all other models is dist(normal), based on the mean and variance entered in es() and var().

corr(*real*) is the correlation between the sensitivity and specificity. The default is corr(0). This option is only needed if the user chooses the bivariate model.

path(*string*) specifies the directory in which to save files created by metasim. This overrides the default of the working directory.

3.3 Example

We illustrate metasim with a systematic review of antibiotic use for the common cold from the Cochrane database of systematic reviews (Arroll and Kenealy 1999). We return to this example in sections 4.3 and 5.3 to illustrate metapow and metapowplot.

Six trials were conducted to compare antibiotics versus placebo for outcome symptoms persisting beyond seven days and labeled as "event" in table 1. A total of 1,147 subjects participated: 664 in the treatment group and 483 in the control group. The trials are summarized in table 1.

Table 1. Six trials included in antibiotics for the common cold and acute purulent rhinitis meta-analysis

Study	Year	a (event/trt)	b (no event/trt)	c (event/ctrl)	d (no event/ctrl)
Herne	1980	7	39	10	12
Hoaglund	1950	39	115	51	104
Kaiser	1996	97	49	94	48
Lexomboon	1971	8	166	4	83
McKerrow	1961	5	10	8	10
Taylor	1977	12	117	3	56

The review concluded that "there was insufficient evidence of benefit to warrant the use of antibiotics" (Arroll and Kenealy 1999). Further trials could be potentially beneficial. A fixed-effects meta-analysis using the inverse-variance method was carried out on the six trials with the OR. The command line is given below, and the results are presented in figure 2.

```
. metan event_t noevent_t event_c noevent_c, or fixedi
> label(namevar=Study, yearvar=Year) textsize(150)
> xlabel(0.125, 0.25, 0.5, 1, 2, 4, 8) scheme(sj)
> title("Forest plot") favours("Favours treatment" # "Favours control")
> xtitle("Odds ratio")
          Study    |   OR    [95% Conf. Interval]    % Weight
--------------------+-------------------------------------------------
Herne (1980)       |  0.215    0.067     0.689          6.87
Hoaglund (1950)    |  0.692    0.422     1.134         38.04
Kaiser (1996)      |  1.011    0.620     1.648         38.88
Lexomboon (1971)   |  1.000    0.293     3.417          6.15
McKerrow (1961)    |  0.625    0.151     2.586          4.61
Taylor (1977)      |  1.915    0.519     7.058          5.46
--------------------+-------------------------------------------------
I-V pooled OR      |  0.796    0.587     1.080         100.00
--------------------+-------------------------------------------------

  Heterogeneity chi-squared =   8.07 (d.f. = 5) p = 0.153
  I-squared (variation in OR attributable to heterogeneity) =  38.0%

  Test of OR=1 : z=   1.47 p = 0.143
```
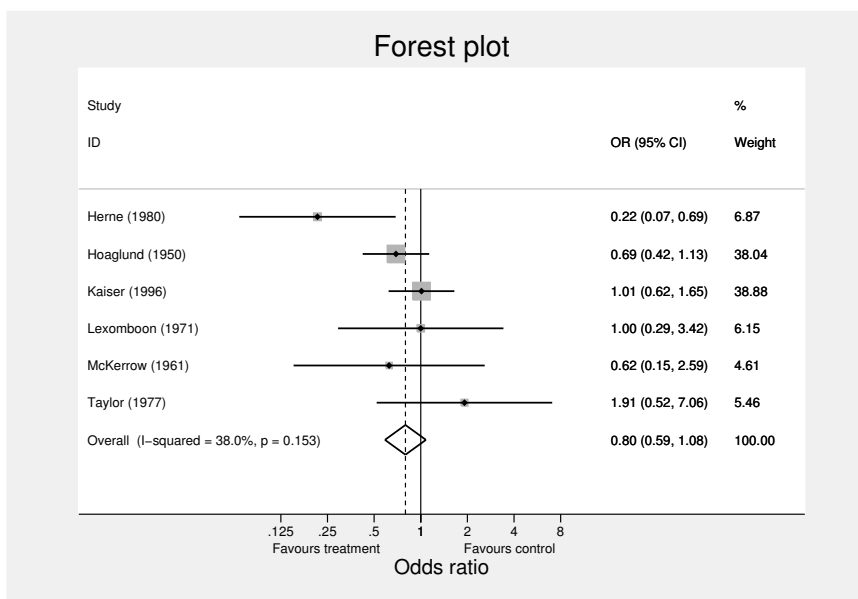
Figure 2. Forest plot of common cold data using fixed-effects meta-analysis with the inverse-variance method

The results suggest a slight treatment benefit, but this is not significant at the 5% level (OR = 0.80, 95% CI: [0.59,1.08]). It is possible that additional information in the form of another trial could lead to this result becoming statistically significant. The command `metasim` allows the user to simulate a new trial based on the above results. By inputting the pooled log OR from the meta-analysis as the estimate, along with the variance of the pooled log OR, users can write the command as follows:

```
. metasim event_t noevent_t event_c noevent_c, es(-0.228) var(0.155) n(100)
> type(clinical) measure(or) model(fixedi)
New study/studies simulated are saved in file called E:\Meta-analysis\
> metapow\temppow
. use temppow, clear
. list
```

	event_t	noeven~t	event_c	noeven~c
1.	21	79	31	69

A new trial has been generated with 100 patients in the treatment arm and 100 patients in the control arm. The trial is saved in the working directory in a file named `temppow`. The file will contain four variables with the same names as those in the current dataset. In this case, these will be `event_t`, `noevent_t`, `event_c`, and `noevent_c`.

4 The metapow command

4.1 Syntax

<u>metapow</u> *varlist*, n(*integer*) nit(*integer*) <u>typ</u>e(clinical | diagnostic)

   pow(*numlist*) $\big[$ <u>measure</u>(or | rr | rd | nostandard | dor | ss)

   <u>inf</u>erence(ciwidth | pvalue | lci | uci) p(*real*) r(*real*) <u>stud</u>ies(*integer*)

   <u>mod</u>el(fixed | fixedi | random | randomi | bivariate) npow(*numlist*) ci(*real*)

   dist(normal | t) ind nip(*integer*) sos(sens | spec) path(*string*)

   level(*integer*) $\big]$

4.2 Options

n(*integer*); see the `metasim` options in section 3.2.

nit(*integer*) is the number of simulations on which the estimated power is based. The larger the number specified, the more accurate the estimate will be, but the longer the analysis will take. nit() is required.

type(clinical | diagnostic); see the `metasim` options in section 3.2.

pow(*numlist*) specifies the value used as a cutoff in determining the power. One or two values may be input. The value represents different things, depending on the option chosen for inference(). pow() is required.

measure(or | rr | rd | nostandard | dor | ss); see the `metasim` options in section 3.2.

inference(ciwidth | pvalue | lci | uci) defines the approach to inference used to calculate power. The default is inference(ciwidth). This counts the number of times that the CI width of the estimate from the updated meta-analysis (that is, with the simulated study included) is less than the specified value. This option can be used regardless of the measure of accuracy. Two other approaches to inference are inference(lci) and inference(uci). These will count the number of times that the lower or upper CI is higher or lower than a given value, respectively. The inference(lci) option can be used regardless of the measure of accuracy. The inference(uci) option is currently only available when working with clinical trial data and not diagnostic data. A final option only available when using clinical trial data is inference(pvalue). This counts the number of times that a *p*-value is significant to a specified level. When you use sensitivity and specificity, two values may be input into pow() for inference(ciwidth) and inference(lci). These will instruct the command to count the number of times that the CI widths for both sensitivity and specificity are less than their respective specified values. Sensitivity must be given first followed by specificity for the calculation to be correct. To use the inference(ciwidth) or inference(lci) option for just sensitivity or just specificity, you should also use the sos() option (described below).

p(*real*); see the `metasim` options in section 3.2.

r(*real*); see the `metasim` options in section 3.2.

studies(*integer*); see the `metasim` options in section 3.2.

model(fixed | fixedi | random | randomi | bivariate); see the `metasim` options in section 3.2.

npow(*numlist*) recalculates the power with a new value for the same `inference()` without having to rerun the whole command. Instead, it uses the data stored in `temppow2` and allows alternative approaches to inference to be explored. This is particularly valuable when the required simulation time is lengthy.

ci(*real*) specifies the width of the CI for the corresponding power estimate. The default is ci(95).

dist(normal | t); see the `metasim` options in section 3.2.

ind instructs the command to calculate the power for the newly simulated study on its own in addition to the newly updated meta-analysis.

nip(*integer*) specifies the number of integration points used for quadrature when the bivariate model is selected. Higher values should result in greater accuracy but typically at the expense of longer execution times (see Harbord and Whiting [2009]).

sos(sens | spec) is used in addition to the `inference()` option and specifies whether inferences are focused on sensitivity or specificity when using `inference(ciwidth)` or `inference(lci)`. The default is sos(sens). If sos() is not specified, then the inferences are based on both sensitivity and specificity, and two values should be entered for pow().

path(*string*); see the `metasim` options in section 3.2.

level(*integer*) specifies the confidence level, as a percentage, for the individual study and pooled CIs. This is the level given in the `metan`, `metandi`, and `midas` commands when called on to meta-analyze the current dataset. The default is level(95).

4.3 Example

The same example described in section 3.3 is used here to demonstrate the command `metapow`. This command allows the user to estimate the power that a new trial of a specified sample size would give to the meta-analysis. In this example, the inference is the p-value. `metapow` is told to estimate the power that a new trial with 100 patients in the treatment arm and 100 patients in the control arm would have at detecting a p-value less than 0.05 in the updated meta-analysis.

```
. metapow event_t noevent_t event_c noevent_c, n(100) type(clinical)
> measure(or) model(fixedi) nit(100) inference(pvalue) pow(0.05)
.................................................................
> ....................
Fixed effect inverse variance-weighted model
Statistic used was odds ratio

n= 100 (in control group)
m= 100 (in treatment group)

Power of meta-analysis is: 31.00 (95% CI: 22.13, 41.03)

Level of significance used to estimate power = 0.05
Simulation estimates are saved in file called E:\Meta-analysis\metapow\temppow2
```

The output from the command describes the type of meta-analysis model specified and the inference used. It also gives the power estimate. In this case, the power estimate is 31.0% (95% CI: [22.1, 41.0]), meaning that the *p*-value was below 0.05 in 31 of the 100 iterations. It is possible to recalculate the power with a different cutoff value without having to rerun the whole analysis. The option npow() can be specified to do this as shown below. Notice that the dots are not displayed, which is because the analysis is not being run. The output also informs the user that the level used to estimate power has changed.

```
. metapow event_t noevent_t event_c noevent_c, n(100) type(clinical)
> measure(or) model(fixedi) nit(100) inference(pvalue) pow(0.05) npow(0.1)

Level used to estimate power has changed
Simulated data has not changed
Fixed effect inverse variance-weighted model
Statistic used was odds ratio

n= 100 (in control group)
m= 100 (in treatment group)

Power of meta-analysis is: 49.00 (95% CI: 38.86, 59.20)

Simulation estimates are saved in file called E:\Meta-analysis\metapow\temppow2
```

metapow stores the estimates from each of the 100 iterations in a file called temppow2. Because the command also calls on metasim, the last newly simulated study will also be saved in a file called temppow. These will both be found within the working directory in Stata.

In this example, only 100 simulations were run, resulting in quite a wide CI for power. This could be reduced by increasing the number of simulations.

5 The metapowplot command

5.1 Syntax

metapowplot *varlist*, start(*#*) stop(*#*) step(*#*) nit(*integer*)

    type(clinical|diagnostic) pow(*numlist*)

    [measure(or|rr|rd|nostandard|dor|ss)

    inference(ciwidth|pvalue|lci|uci) p(*real*) r(*real*) studies(*integer*)

    model(fixed|fixedi|random|randomi|bivariate) npow(*numlist*) ci(*real*)

    dist(normal|t) ind nip(*integer*) sos(sens|spec) path(*string*)

    graph(lowess|connected|overlay) noci regraph level(*integer*)]

5.2 Options

start(*#*) is the smallest total sample size of a new study for which the user wishes to calculate a power value. start() is required.

stop(*#*) is the largest total sample size of a new study for which the user wishes to calculate a power value. stop() is required.

step(*#*) is the step size to be used within the range of total sample sizes specified by start() and stop(). A step size of 10 between the range of 10 to 30 would mean that the power would be estimated for sample sizes of 10, 20, and 30. step() is required.

nit(*integer*); see the metapow options in section 4.2.

type(clinical|diagnostic); see the metasim options in section 3.2.

pow(*numlist*); see the metapow options in section 4.2.

measure(or|rr|rd|nostandard|dor|ss); see the metasim options in section 3.2.

inference(ciwidth|pvalue|lci|uci); see the metasim options in section 3.2.

p(*real*); see the metasim options in section 3.2.

r(*real*); see the metasim options in section 3.2.

studies(*integer*); see the metasim options in section 3.2.

model(fixed|fixedi|random|randomi|bivariate); see the metasim options in section 3.2.

npow(*numlist*); see the metapow options in section 4.2.

ci(*real*); see the metapow options in section 4.2.

dist(normal|t); see the metasim options in section 3.2.

ind; see the `metapow` options in section 4.2.

nip(*integer*); see the `metapow` options in section 4.2.

sos(sens | spec); see the `metapow` options in section 4.2.

path(*string*); see the `metasim` options in section 3.2.

graph(lowess | connected | overlay) allows the user to choose the type of line used to connect the specific estimates of power at the specified sample sizes. The default is graph(connected), which plots each point and connects them with a line. The other options are a lowess plot, which plots a smoothed line to the specific points, and an overlay plot, which plots both the points and the lowess curve. Because power is estimated through simulation, there is sampling error in each estimate that will decrease with the number of simulations specified (but also increase evaluation time). Thus smoothing may be desirable if several different but inaccurate estimates are considered. The lowess line should be similar to the connected option for larger simulations.

noci prevents the command from plotting CIs (indicating the sampling error in the estimation of power at specified sample sizes) on the graph.

regraph allows the user to regraph the power curves with alternative graph options without having to rerun the simulations for the specified range of sample sizes.

level(*integer*); see the `metapow` options in section 4.2.

5.3   Example

The command `metapowplot` is used to calculate the power value at various sample sizes by calling on `metapow`. The command then plots the power values against sample size. In the command below, the range of sample sizes has been specified as 100 to 1,000 with steps of 100; the results are shown in figure 3. All other options remain the same as those in section 4.3.

```
. metapowplot event_t noevent_t event_c noevent_c, start(100) step(100)
> stop(1000) type(clinical) measure(or) model(fixedi) nit(100)
> inference(pvalue) pow(0.05)

Sample size

t =      100       Treatment/Control = 50/50
t =      200       Treatment/Control = 100/100
t =      300       Treatment/Control = 150/150
t =      400       Treatment/Control = 200/200
t =      500       Treatment/Control = 250/250
t =      600       Treatment/Control = 300/300
t =      700       Treatment/Control = 350/350
t =      800       Treatment/Control = 400/400
t =      900       Treatment/Control = 450/450
t =     1000       Treatment/Control = 500/500
```

```
Fixed effect inverse variance-weighted model
Statistic used was odds ratio
Level of significance used to estimate power = 0.05
Power estimates used to plot the graph are saved in file called E:\Meta-analysis\
> metapow\temppow3
```
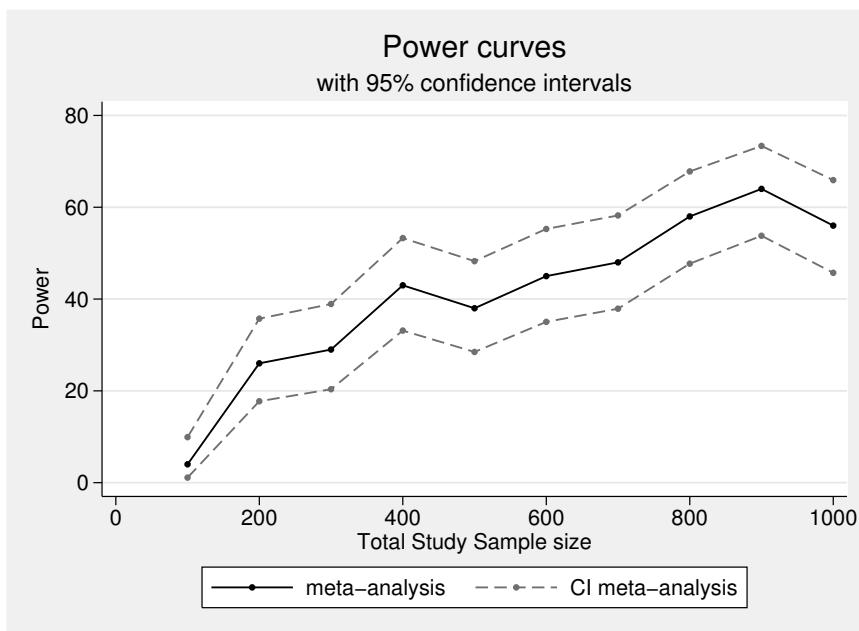

Figure 3. Power curve for common cold data based on the OR using a fixed-effects model with the inverse-variance method

metapowplot has stored the power values and corresponding sample sizes in a file called temppow3. Because the command calls on both metapow and metasim, the estimates from the last sample size are stored in temppow2, and the final newly simulated study is stored in temppow. All these files can be found in the working directory.

The output describes the options chosen by the user. Figure 3 shows the power curve generated by metapowplot. The power is estimated to reach 60% with a total sample size of 800: 400 patients in the treatment arm and 400 patients in the control arm. This implies that when updated with more information, the current meta-analysis from the Cochrane database could provide significant evidence to suggest a benefit in the use of antibiotics for the common cold.

If you are designing a new trial, we would recommend running a minimum of 1,000 simulations at each sample size and perhaps over a narrower, targeted range of sample sizes of interest.

5.4   Diagnostic example

This example focuses on the diagnostic test accuracy options within the commands. A meta-analysis was carried out in 1999 to assess the diagnostic value of the digital rectal examination (DRE) in detecting prostate cancer (Hoogendam, Buntinx, and de Vet 1999). Studies were included if they compared DRE with biopsy or surgery as the reference standard. A total of 14 studies met the inclusion criteria, giving a total of 21,839 patients. Table 2 gives the results from each study.

Table 2. Fourteen studies included in DRE as screening test for prostate cancer

Study	Year	TP	FP	FN	TN
Kirky	1994	8	6	6	541
Vihko	1985	6	21	3	741
Chodak	1989	32	112	13	1974
Ciatto	1994	17	8	9	1391
Lee	1989	10	19	12	743
Pode	1995	22	93	9	876
Dalkin	1993	9	33	15	695
Palken	1991	17	28	6	264
Teillac	1990	8	18	10	546
Catalona	1994	146	836	118	5530
Menor	1990	59	48	16	1389
Richie	1994	16	194	8	426
Gustafsson	1992	42	153	23	1564
Littrup	1994	77	287	95	2471

A separate DerSimonian and Laird random-effects meta-analysis of sensitivity and specificity was carried out on the 14 studies. Figure 4 gives the results of the random-effects meta-analysis of sensitivity. The results give a pooled estimate for sensitivity of 0.60 (95% CI: [0.53, 0.67]). This suggests that the test correctly identifies only 60% of the diseased patients. The other 40% would be given false negative results. The results of the random-effects meta-analysis of specificity are shown in figure 5. The pooled estimate for specificity was 0.95 (95% CI: [0.92, 0.96]). This suggests that 95% of the healthy patients are correctly identified by the test. This result is fairly good because it means that only 5% of the healthy patients would receive false positive results.
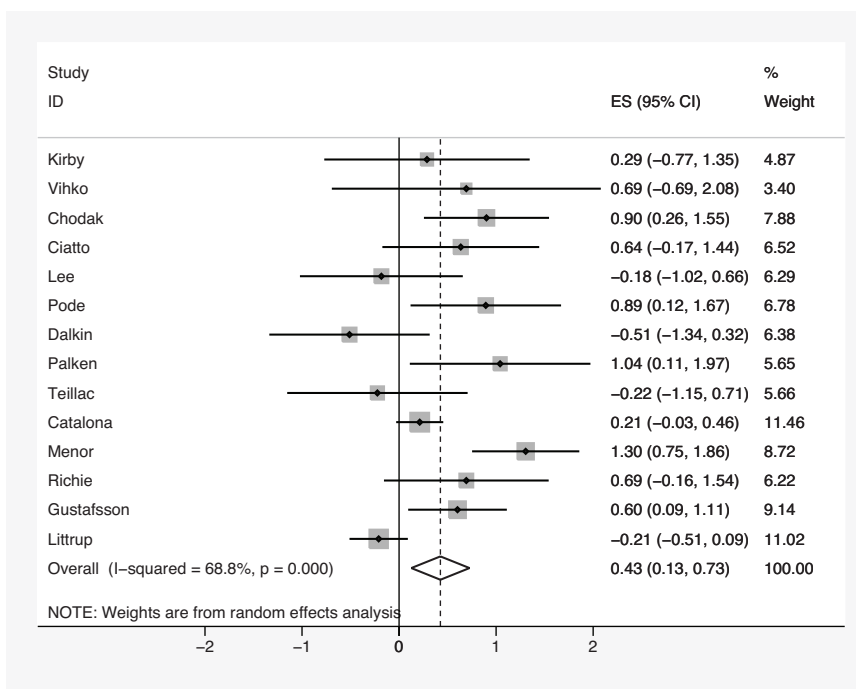
Figure 4. Forest plot of prostate data using DerSimonian and Laird random-effects meta-analysis of logit sensitivity
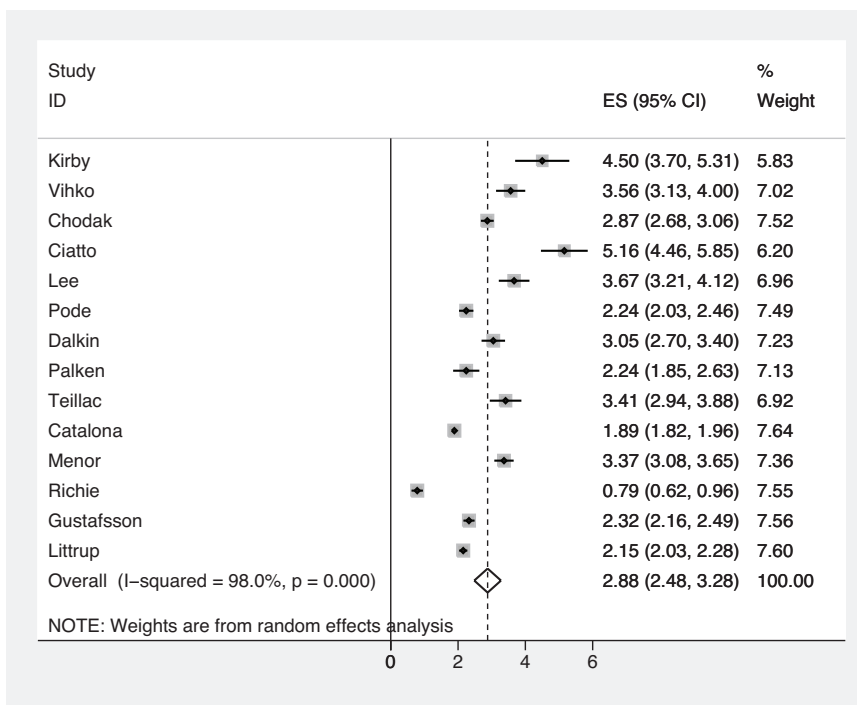
Figure 5. Forest plot of prostate data using DerSimonian and Laird random-effects meta-analysis of logit specificity

The command line given below estimates power values for sample sizes ranging from 100 to 1,000 in steps of 100 based on the lower confidence interval value for the pooled sensitivity estimate only. The cutoff for this has been set to 0.53, which is the same as the lower confidence interval value in the current meta-analysis for sensitivity (see figure 4). `metapowplot` will count how many times the lower confidence interval value for the pooled sensitivity is greater than or equal to 0.53 and base the power value on this.

```
. metapowplot TP FP FN TN, start(100) step(100) stop(1000) type(diagnostic)
> measure(ss) model(randomi) nit(200) sos(sens) inference(lci) pow(0.53)
Sample size
t =     100     Diseased/Healthy = 50/50
t =     200     Diseased/Healthy = 100/100
t =     300     Diseased/Healthy = 150/150
t =     400     Diseased/Healthy = 200/200
t =     500     Diseased/Healthy = 250/250
t =     600     Diseased/Healthy = 300/300
t =     700     Diseased/Healthy = 350/350
t =     800     Diseased/Healthy = 400/400
t =     900     Diseased/Healthy = 450/450
t =    1000     Diseased/Healthy = 500/500

Random effects model with inverse variance-weighted estimates of heterogeneity
Statistics used were sensitivity and specificity
Lower confidence interval value for sens used to estimate power = 0.53
Power estimates used to plot the graph are saved in file called E:\Meta-analysis\
> metapow\temppow3
```
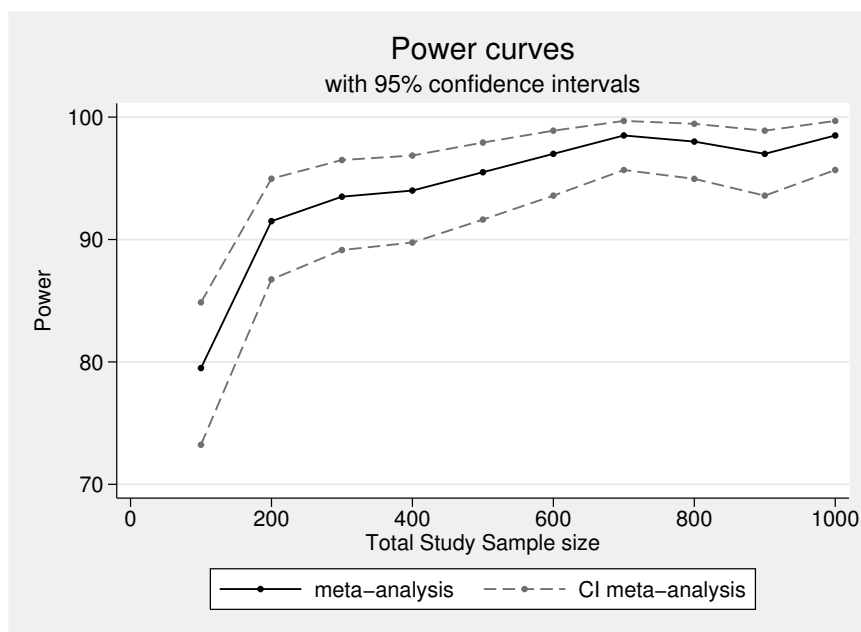

Figure 6. Power curve for prostate data based on sensitivity using DerSimonian and Laird random-effects model

Figure 6 shows the power curve obtained from the above command line. The power reaches about 98% for a total sample size of 1,000. This means that when a study with 500 diseased patients and 500 healthy patients was added to the current meta-analysis for sensitivity, the lower CI value for the pooled sensitivity was greater than or equal to 0.53 in about 98 of the 100 iterations.

6   Other uses

An intuitive way to visualize this process is to plot all the results of the individual simulations, at a specified sample size, on an extended funnel plot (Langan et al. 2012; Crowther, Langan, and Sutton 2012). Extended funnel plots illustrate how the conclusions of a meta-analysis would be impacted by the addition of a single new trial across a range of effect estimates and standard errors. By directly overlaying the simulation results at a specific sample size, stored in `temppow2.dta`, we can draw direct conclusions about the area where a new study would likely lie and its impact on hypothesis tests.

In figure 7, we overlay the simulated individual studies from the example in section 4.3. The majority of points lies in the region of the plot where a new study, when added to the existing meta-analysis, would produce a statistically significant result, with the updated effect estimate and 95% CI less than the null. This process can be repeated as desired for different sample sizes. In this example, we can directly relate the 28% power to 28 of the 100 simulated studies lying in the left-hand region of the plot, indicating a change in conclusions for the updated meta-analysis.

```
. merge 1:1 _n using "temppow2.dta", nogen noreport
. metan event_t noevent_t event_c noevent_c, or fixedi nograph
  (output omitted)
. gen logor=log(_ES)
(94 missing values generated)
. gen t1 = log(indes)
. extfunnel logor _selogES, fixedi eform
> xlabel(0.1 0.2 0.5 1 2 5, format(%2.1f)) yrange(0 1)
> addplot(scatter indse_es indes, msize(tiny) msym(T) mcol(black) xscale(log))
> ylabel(,format(%2.1f)) sumd sumdpos(0.9) pred
> legend(order(1 "Non-sig. effect (5% level)" 2 "Sig. effect > NULL (5% level)"
> 3 "Sig. effect < NULL (5% level)" 4 "Prediction interval" 6 "Null effect"
> 7 "Pooled effect" 8 "Original studies" 9 "Simulated studies"))
Original meta-analysis results:
          Study      |    ES    [95% Conf. Interval]    % Weight
--------------------+-----------------------------------------------
1                   |  0.215    0.067    0.689          6.87
2                   |  0.692    0.422    1.134         38.04
3                   |  1.011    0.620    1.648         38.88
4                   |  1.000    0.293    3.417          6.15
5                   |  0.625    0.151    2.586          4.61
6                   |  1.915    0.519    7.058          5.46
--------------------+-----------------------------------------------
I-V pooled ES       |  0.796    0.587    1.080        100.00
--------------------+-----------------------------------------------

  Heterogeneity chi-squared =   8.07 (d.f. = 5) p = 0.153
  I-squared (variation in ES attributable to heterogeneity) =  38.0%

  Test of ES=1 : z=   1.47 p = 0.143
Building graph:
```
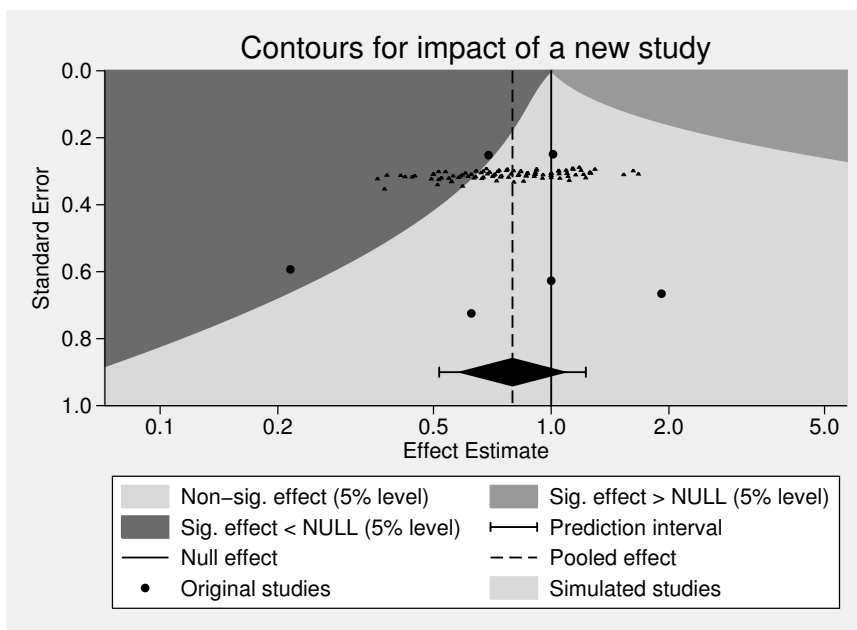
Figure 7. Extended funnel plot with simulated studies overlaid

7   Discussion

We hope the commands will be useful to 1) trialists who want to assess the potential impact new trials will have on the overall evidence base and those involved in funding new trials; and 2) meta-analysts who want to assess the robustness of the current meta-analysis to the inclusion of future data.

Finally, we thought it may be helpful to outline ongoing and potential future work. We have created a prototype set of commands that conducts the same calculations as the commands described here but that uses a Bayesian approach to all meta-analyses estimation. This is done through the use of the WinBUGS software, which links with Stata through a previously written command (Thompson, Palmer, and Moreno 2006). We hope to develop these to a point where they can be released in the future because a Bayesian approach to meta-analysis offers several advantages, as described elsewhere (Sutton et al. 2007).

A further Stata command, with the specific purpose of prioritizing a portfolio of meta-analyses for updating and which adapts much of the methodology described herein, is also very near completion.

Finally, others have extended the approaches described here to the context of cluster randomized controlled trials (Rotondi and Donner 2012) and written software in R to implement them; we hope this extension of the methodology can also be coded in a Stata command.

8   Acknowledgments

9   References

Arroll, B., and T. Kenealy. 1999. Antibiotics versus placebo in the common cold (Cochrane Review). *Cochrane Library Oxford* Issue 2.

Cooper, N. J., D. R. Jones, and A. J. Sutton. 2005. The use of systematic reviews when designing studies. *Clinical Trials* 2: 260–264.

Crowther, M. J., D. Langan, and A. J. Sutton. 2012. Graphical augmentations to the funnel plot to assess the impact of a new study on an existing meta-analysis. *Stata Journal* 12: 605–622.

Dwamena, B. 2007. midas: Stata module for meta-analytical integration of diagnostic test accuracy studies. Statistical Software Components S456880, Department of Economics, Boston College. http://ideas.repec.org/c/boc/bocode/s456880.html.

Feiveson, A. H. 2002. Power by simulation. *Stata Journal* 2: 107–124.

Goudie, A. C., A. J. Sutton, D. R. Jones, and A. Donald. 2010. Empirical assessment suggests that existing evidence could be used more fully in designing randomized controlled trials. *Journal of Clinical Epidemiology* 63: 983–991.

Harbord, R. M., and P. Whiting. 2009. metandi: Meta-analysis of diagnostic accuracy using hierarchical logistic regression. *Stata Journal* 9: 211–229.

Harris, R. J., M. J. Bradburn, J. J. Deeks, R. M. Harbord, D. G. Altman, and J. A. C. Sterne. 2008. metan: Fixed- and random-effects meta-analysis. *Stata Journal* 8: 3–28.

Hinchliffe, S. R., M. J. Crowther, R. S. Phillips, and A. J. Sutton. 2013. Using meta-analysis to inform the design of subsequent studies of diagnostic test accuracy. *Research Synthesis Methods* 4: 156–168.

Hoogendam, A., F. Buntinx, and H. C. de Vet. 1999. The diagnostic value of digital rectal examination in primary care screening for prostate cancer: A meta-analysis. *Family Practice* 16: 621–626.

Langan, D., J. P. T. Higgins, W. Gregory, and A. J. Sutton. 2012. Graphical augmentations to the funnel plot assess the impact of additional evidence on a meta-analysis. *Journal of Clinical Epidemiology* 65: 511–519.

Rabe-Hesketh, S., A. Skrondal, and A. Pickles. 2005. Maximum likelihood estimation of limited and discrete dependent variable models with nested random effects. *Journal of Econometrics* 128: 301–323.

Rotondi, M., and A. Donner. 2012. Sample size estimation in cluster randomized trials: An evidence-based perspective. *Computational Statistics and Data Analysis* 56: 1174–1187.

Sutton, A. J., N. J. Cooper, and D. R. Jones. 2009a. Evidence synthesis as the key to more coherent and efficient research. *BMC Medical Research Methodology* 9: 29.

Sutton, A. J., N. J. Cooper, D. R. Jones, P. C. Lambert, J. R. Thompson, and K. R. Abrams. 2007. Evidence-based sample size calculations based upon updated meta-analysis. *Statistics in Medicine* 26: 2479–2500.

Sutton, A. J., S. Donegan, Y. Takwoingi, P. Garner, C. Gamble, and A. Donald. 2009b. An encouraging assessment of methods to inform priorities for updating systematic reviews. *Journal of Clinical Epidemiology* 62: 241–251.

Thompson, J., T. Palmer, and S. Moreno. 2006. Bayesian analysis in Stata with Win-BUGS. *Stata Journal* 6: 530–549.

About the authors

Michael Crowther is a research associate at the University of Leicester, UK. His main area of research is the joint modeling of longitudinal and survival data.

Sally Hinchliffe is a PhD student at the University of Leicester, UK. She is currently working on developing methodology for application in competing risks.

Alison Donald worked on this project as part of her Master of Science in medical statistics dissertation project at the University of Leicester.

Alex Sutton is a professor of medical statistics at the University of Leicester, UK. He has a long-standing research interest in meta-analysis and evidence synthesis.