Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. *http://www.jstatsoft.org/*

Simulation-Based Power Calculations for Mixed Effects Modelling: ipdpower in Stata

Evangelos Kontopantelis
NIHR School for Primary Care Research
University of Manchester

David A Springate
Institute of Population Health
University of Manchester

Rosa Parisi
Manchester Pharmacy School
University of Manchester

David Reeves
Institute of Population Health
University of Manchester

Abstract

Simulations are a practical and reliable approach to power calculations, especially for multi-level mixed effects models where the analytic solutions can be very complex. In addition, power calculations are model-specific and multi-level mixed effects models are defined by a plethora of parameters. In other words, model variations in this context are numerous and so are the tailored algebraic calculations. This article describes **ipdpower** in Stata, a new simulations-based command that calculates power for mixed effects two-level data structures. Although the command was developed having individual patient data meta-analyses and primary care databases analyses in mind, where patients are nested within studies and general practices respectively, the methods apply to any two-level structure.

Keywords: Stata, **ipdpower**, power, coverage, meta analysis, multi level, mixed effects, random effects, individual patient data, IPD, primary care databases, PCD.

1. Introduction

The size of primary care databases (PCDs) allows for investigations that cannot normally be undertaken in much smaller randomised controlled trials (RCTs), such as the moderating effect of a patient characteristic on the effect of an intervention. However, researchers quite often underestimate the numbers needed to detect such effects and assume that the size of the database alone guarantees adequate power for any type of investigation. The essential

structure of a PCD dataset closely resembles that of an individual patient data (IPD) meta-analysis, with patients nested (clustered) in general practices in the former and in studies in the latter, thus the same modelling approaches and power analysis considerations are applicable to both. In individual patient data meta-analyses, the quantity of the collected data from numerous RCTs sometimes deceives researchers with regards to the power in their analysis and can lead to the assumption that a power calculation is not necessary. However, even though power calculations are quite often needed in these contexts, especially for detecting a moderating effect, available software options are not user friendly or can only cope with simple models (e.g., without random effects for the higher level clusters). Hence, it is not uncommon for researchers to use a rough 'four-times as many patients needed as for the main effect' over-simplification to estimate power (McClelland and Judd 1993).

More recently, an algebraic approximation approach, which uses study summary statistics, was developed by Kovalchik and Cumberland (2012) and implemented in R (Kovalchik 2013; R Development Core Team 2013) for individual patient data meta-analysis. Overall, Kovalchik and Cumberland found that their computationally cheap method performed well in simulations although it is constrained by distributional assumptions and performance deteriorated in some scenarios. Although this approach can be very useful, it is limited to a randomised controlled trial setting with a binary treatment, a patient-level covariate and a binary or continuous outcome. In addition, random effects for the intercept and intervention were modelled but other random effects (e.g., for the covariate) were not considered.

Using simulations to calculate power is a well-known but computationally expensive approach (Feiveson 2002), but with the availability of powerful computers, simulations can be a very useful alternative when it comes to complex study designs for which power equations are unavailable or prohibitively complex (Arnold, Hogan, Colford, and Hubbard 2011). The **ipdpower** command for Stata (StataCorp. 2011) provides a simple but flexible framework for simulation-based power calculations. Various random effects at the cluster (e.g., practice, study) level can be hypothesised, the outcome can be continuous, binary or count data, covariates and moderators (i.e., interactions) can be modelled at the cluster- or patient-level, exposure can be binary (e.g., intervention) or continuous (as is often the case in observational studies) and distributions for the random effects and continuous variables can be simulated to be non-normal. Besides power, the command provides additional information that can be useful when designing a study, for example coverage (type I error) which might depart from the nominal level when the data structure is complex.

2. Methods

The command proceeds in two steps, within each simulation iteration. First, it generates a dataset, defining the outcome according to the specified coefficients for the intercept (constant), the exposure (intervention), the covariate and the exposure-covariate interaction (moderator effect). Second, it uses regression modelling to calculate model fit statistics, power and coverage for the dataset. Information is then aggregated across all simulated datasets. Power indicates the percentage of iterations in which a model coefficient was found to be statistically significant and in the hypothesised direction. Coverage indicates the percentage of confidence intervals around the coefficient that include the true value and should correspond to the hypothesised α level. Binomial proportion confidence intervals using the **cii** command are calculated for both power and coverage.

2.1. Dataset generation

For a continuous outcome, each dataset is generated using the following set of equations:

$$Y_{ij} = \beta_{0j} + \beta_{1j} group_{ij} + \beta_{2j} X_{ij} + \beta_{3j} group_{ij} * X_{ij} + \epsilon_{ij} \tag{1a}$$

$$\begin{aligned} \beta_{0j} &= \gamma_0 + u_{0j} \\ \beta_{1j} &= \gamma_1 + u_{1j} \\ \beta_{2j} &= \gamma_2 + u_{2j} \\ \beta_{3j} &= \gamma_3 + u_{3j} \end{aligned} \tag{2}$$

and when assuming a normal distribution for the error and random effects

$$\begin{aligned} \epsilon_{ij} &\sim N(0, \sigma_j^2) \\ u_{0j} &\sim N(0, \tau_0^2) \\ u_{1j} &\sim N(0, \tau_1^2) \\ u_{2j} &\sim N(0, \tau_2^2) \\ u_{3j} &\sim N(0, \tau_3^2) \end{aligned} \tag{3}$$

where
i the patient
j the the cluster (e.g., study)
Y_{ij} the outcome for patient i in cluster j
$group_{ij}$ exposure for patient i in cluster j
X_{ij} the covariate (e.g., baseline level) for patient i in cluster j
$group_{ij} * X_{ij}$ the exposure-covariate interaction (moderator) for patient i in cluster j
β_{0j} the intercept for cluster j
γ_0 the mean common intercept
β_{1j} the exposure effect for cluster j
γ_1 the mean exposure effect
β_{2j} the covariate effect for cluster j
γ_2 the mean covariate effect
β_{3j} the interaction effect for cluster j
γ_3 the mean interaction effect
u_{0j} the random intercept for cluster j
u_{1j} the random exposure effect for cluster j
u_{2j} the random covariate effect for cluster j
u_{3j} the random interaction effect for cluster j
${\tau_0}^2$ the between-cluster variance for the intercept
${\tau_1}^2$ the between-cluster variance for the exposure effect
${\tau_2}^2$ the between-cluster variance for the covariate effect
${\tau_3}^2$ the between-cluster variance for the interaction effect
ϵ_{ij} the error term for patient i in cluster j
σ_j^2 the within-cluster variance for cluster j

If the outcome is dichotomous equation (1a) becomes:

$$\ln\left(\frac{p_{ij}}{1-p_{ij}}\right) = \beta_{0j} + \beta_{1j}group_{ij} + \beta_{2j}X_{ij} + \beta_{3j}group_{ij} * X_{ij} \tag{1b}$$

where p_{ij} the probability that the outcome is 1 for patient i in cluster j and coefficients now correspond to log odds (β_0) and log odds ratios (β_1,β_2 and β_3).

And for a count outcome, equation (1a) becomes:

$$\ln(E(Y_{ij})) = \beta_{0j} + \beta_{1j}group_{ij} + \beta_{2j}X_{ij} + \beta_{3j}group_{ij} * X_{ij} \tag{1c}$$

where $E(Y_{ij})$ the expectation of Y_{ij} and the coefficients correspond to log incidence (β_0) and log incidence rates(β_1,β_2 and β_3).

However, we also implemented two alternative skew-normal distributions for the random effects and residual error using methods described by Ramberg, Dudewicz, Tadikamalla, and Mykytka (1979). Besides normal distributions (skewness=0; kurtosis=3), **ipdpower** can model moderate-skew (skewness=1; kurtosis=4) and extreme-skew (skewness=2; kurtosis=9) distributions with the same means and variances as described in equation (3). Although non-normal distributions for the random effects have not been found to notably affect the performance of two-stage meta-analysis methods (Kontopantelis and Reeves 2012), deviations from normality are not necessarily insignificant in this context.

ipdpower allows various random effects components (intercept, exposure, covariate and interaction) to be incorporated in each dataset. In meta-analysis nomenclature, the variance associated with random effects is described as heterogeneity of the exposure (treatment) effect in two-stage meta-analyses (stage 1: obtain or calculate study results; stage 2: calculate overall effect using an appropriate method). When individual patient data are available, meta-analysts can choose one of many mixed effects regression models to combine information and model various random effects in a single stage (Kontopantelis and Reeves 2013), which is considered the best approach to meta-analysis (Mathew and Nordstrom 2010). Usually heterogeneity is quantified with I^2 (Higgins and Thompson 2002) or H^2 (Mittlbock and Heinzl 2006), using the within (σ^2) and between-study (${\tau_1}^2$) variance estimates (assuming within-variance is similar across clusters). Since the error ϵ (defined using variance σ_j^2) is only meaningful in OLS regressions, these heterogeneity measures are only relevant when the outcome is continuous in IPD meta-analysis (in Stata, σ_j^2 is fixed to $\pi^2/3$ for logistic regressions). For this reason **ipdpower** accepts between-cluster variance inputs (${\tau_0}^2$, ${\tau_1}^2$, ${\tau_2}^2$ and ${\tau_3}^2$) to define the random effects for each cluster. However, when the outcome is continuous and users wish to utilise I^2 as a starting point to define model heterogeneity, solving for τ^2 returns (Higgins and Thompson 2002):

$${\tau_i}^2 = \frac{\sigma^2 {I_i}^2}{100 - {I_i}^2} \tag{4}$$

Similarly, ${\tau_i}^2 = ({H_i}^2 - 1)\sigma^2$, where $i \in \{0, 1, 2, 3\}$, and the command also provides I^2 and H^2 as outputs for users to double-check their calculations.

2.2. Regression modelling

ipdpower provides seven different regression modelling options, consistent across the three possible outcome types, to account for the various levels of complexity in the generated datasets. The simplest analysis approach, `model(1)`, does not account for clustering, assumes no random effects and employs commands **regress**, **logit** or **poisson** for continuous, binary or count outcomes respectively. A more advanced approach (`model(2)`), declares the data as clustered using **xtset** and analyses with **xtreg**, **xtlogit** or **xtpoisson**. With this family of models only a random effects component for the intercept (${\tau_0}^2 > 0$) is considered and estimated. More advanced modelling options, which allow consideration of more random effects components, have also been implemented. However, these models which use **xtmixed**, **xtmelogit** and **xtmepoisson** (renamed in Stata v13), can be computationally expensive and do not always converge. These modelling approaches have been described in the IPD meta-analysis context (Higgins, Whitehead, Turner, Omar, and Thompson 2001; Turner, Omar, Yang, Goldstein, and Thompson 2000; Whitehead 2002; Kontopantelis and Reeves 2013). The simplest of them, `model(3)`, assumes a fixed common intercept, random exposure effects (${\tau_1}^2 > 0$) and fixed effect for the covariate. The recommended model for meta-analyses (Whitehead 2002), `model(4)`, assumes fixed study specific intercepts, random exposure effects (${\tau_1}^2 > 0$) and fixed study specific effects for the covariate. `model(5)` assumes random study intercepts (${\tau_0}^2 > 0$), random exposure effects (${\tau_1}^2 > 0$) and fixed study specific effect for the covariate. `model(6)` assumes random study intercepts (${\tau_0}^2 > 0$), random exposure effects (${\tau_1}^2 > 0$) and random effects for the covariate (${\tau_2}^2 > 0$). `model(7)` assumes random study intercepts (${\tau_0}^2 > 0$), random exposure effects (${\tau_1}^2 > 0$), random effects for the covariate (${\tau_2}^2 > 0$) and random effects for the interaction (${\tau_3}^2 > 0$). The more complicated the model the more likely analysis will fail to converge in a particular dataset, and this is often the case for the last three models - especially the last one.

Although fixed effect models are widely used in two-stage meta-analyses, even when heterogeneity is not zero (Kontopantelis, Springate, and Reeves 2013), accounting for even low levels of between-cluster variability is a more conservative approach (Hunter and Schmidt 2000). When a fixed effects model is incorrectly assumed, both coverage and power deteriorate as true heterogeneity increases (Brockwell and Gordon 2001; Kontopantelis and Reeves 2012). Analogously, for patient data analyses, we would expect poor fit from `model(1)`, the fixed effect approach, in the presence of heterogeneity. More generally, the closer the modelling assumptions to the true data structure, the better the expected performance. However, as explained previously, the more complex models come with practical limitations so users must make the choice that best suits their needs in this trade-off between feasibility and model correctness. From this point of view, the usability of **ipdpower** is not limited to power calculations for a moderator effect, but it can be used to evaluate the overall performance of current modelling approaches in various simulated scenarios. For example, to compare `model(4)`, the preferred IPD meta-analysis approach, to the simpler `model(3)`.

3. The ipdpower command

3.1. Syntax

`ipdpower, sn(#) ssl(#) ssh(#) b0(#) b1(#) b2(#) b3(#) [minsh(#) hpoisson icluster outc(#) cb(#) cexp cexpd(#) errsd(#) sderrsd(#) derr(#) ccovd(#) bcov bcb(#) slcov tsq0(#) tsq1(#) tsq2(#) tsq3(#) dtp0(#) dtp1(#) dtp2(#) dtp3(#) covmat(`*name*`) missp(#) mar(#) mnar(#) minum(#) mipmm(#) model(#) clvl(#) seed(#) nskip dnorm xnodts di moreon]`

3.2. Required

`sn(#)` Number (*integer*) of simulations to execute. At least 1000 are recommended for relatively narrow confidence intervals.

`ssl(#)` Total number (*integer*) of patients across all higher level units (clusters).

`ssh(#)` Number (*integer*) of higher level units (e.g., studies, general practices etc).

`b0(#)` Coefficient (*real*) for the intercept (constant) of regression model. For logistic and Poisson regression log odds and log incidence rates are expected respectively. The coefficient can be zero.

`b1(#)` Coefficient (*real*) for the exposure variable (e.g., intervention: treatment vs no treatment) of the regression model. For logistic and Poisson regression log odds ratios and log incidence rate ratios are expected respectively. The coefficient can be zero.

`b2(#)` Coefficient (*real*) for the covariate variable (e.g., age) of the regression model. It can be continuous (default) or binary (`bcov` option), and patient-level (default) or study-level (`slcov` option). For logistic and Poisson regression log odds ratios and log incidence rate ratios are expected respectively. The coefficient can be zero. Note that the covariate, when continuous, is always assumed to be standardised (mean=0 and sd=1) since interactions are included in the models. Users need to take that into consideration when deciding on b2.

`b3(#)` Coefficient (*real*) for the exposure-covariate interaction variable. The command can automatically handle a binary by continuous, binary by binary or continuous by continuous interaction term. For logistic and Poisson regression log odds ratios and log incidence rate ratios are expected respectively. The coefficient can be zero. Note that the covariate, when continuous, is always assumed to be standardised (mean=0 and sd=1) to allow for meaningful estimates. Users need to take that into consideration when deciding on b3.

3.3. Optional

Data Structure

`minsh(#)` Minimum number (*integer*) of patients in a higher level unit. The default is 50 since this is usually the threshold above which the effort required to obtain individual patient data for meta-analysis is justified. The command uses the numbers provided with `ssl(#)`, `ssh(#)` and `minsh(#)` to draw sizes for the higher level units from a uniform distribution. If the average size for the higher level unit is smaller than the minimum number of patients the

command will return an error.

`hpoisson` Inform the command that the higher level unit sizes will not be drawn from a uniform but a Poisson distribution with mean=`ssl(#)`/`ssh(#)`. This approach provides cluster sizes that are much more similar in size. Cannot be used with option `minsh(#)`.

`icluster` Inform the command that the exposure is clustered at the higher level units (e.g., cluster-RCT). Can only be selected when exposure is binary and with balanced designs (i.e., cannot be used with `cexp` or `cb(#)`). Clusters assigned an odd identifier are assumed to include controls and clusters assigned even identifiers are assumed to include the intervention cases (i.e., if an odd number of higher level units is simulated with `icluster` there will be an additional cluster of controls).

`outc(#)` Type of outcome: 0=continuous (default); 1=dichotomous; 2=count. The model for a continuous outcome is $y = b0 + b1 * grp + b2 * xcovar + b3 * xcovar * grp + u0 + u1 * grp + u2 * xcovar + u3 * xcovar * grp + errx$, where *grp* the exposure variable, *xcovar* the covariate, *xcovar*grp* their interaction, *u0-u3* the random effects components and *errx* the residual errors. For a binary outcome the model is $y = uniform()<invlogit(b0 + b1 * grp + b2 * xcovar + b3 * xcovar * grp + u0 + u1 * grp + u2 * xcovar + u3 * xcovar * grp)$ and for a count outcome it is $y = rpoisson(exp(b0 + b1 * grp + b2 * xcovar + b3 * xcovar * grp + u0 + u1 * grp + u2 * xcovar + u3 * xcovar * grp))$. Negative binomial models were considered as an alternative for count data but they were thought to be too complex to be of much practical use in the assumption-laden context of power calculations. Note that residual errors can only be directly controlled in the OLS regression model.

`cb(#)` Probability (*real*) for patient membership to the exposure group (grp=1), when exposure is binary. The default is 0.5 for a balanced design.

`cexp` Inform the command that exposure is continuous (standardised, i.e., mean=0 and sd=1) rather than binary (the default).

`cexpd(#)` Distribution for continuous exposure: 0=normal; 1=moderate skew; 2=extreme skew. Normal distribution for the exposure (skew=0, kurtotis=3) is the default. Moderate (skew=1, kurtotis=4) and extreme skewness (skew=2, kurtotis=9) are implemented using Ramberg *et al.* (1979) method. The distribution for the exposure will not affect the distribution of the outcome much unless b1 is reasonably large. Note that the exposure, when continuous, is always assumed to be standardised (mean=0 and sd=1) since interactions are included in the models. Users need to take that into consideration when deciding on b1 and b3.

`errsd(#)` For continuous outcome only, standard deviation for the residual error (*real*). The default is 1. This value, combined with the model coefficients, will affect the model fit; e.g., a large value will drive down the average adjusted R^2. It also affects model heterogeneity since this is effectively the sd for the outcome within the higher level unit (e.g., within-study variability).

`sderrsd(#)` For continuous outcome only, standard deviation for the standard deviation of the residual error (*real*). In other words, it allows the residual error to vary across higher-level units, which might be a more realistic modelling strategy. The default is 0, not allowing variation which complies with modelling and heterogeneity assumptions (e.g., pooled within-variance is used for heterogeneity calculations).

`derr(#)` For continuous outcome only, distribution for errors and hence outcome: 0=normal; 1=moderate skew; 2=extreme skew. Normal distribution for the error (skew=0, kurtotis=3)

is the default. Moderate (skew=1, kurtotis=4) and extreme skewness (skew=2, kurtotis=9) are implemented using Ramberg *et al.* (1979) method. The distribution for the errors will affect the distribution of the outcome and the larger the modelled errors the more similar the two distributions.

`ccovd(#)` Distribution for continuous covariate: 0=normal; 1=moderate skew; 2=extreme skew. Normal distribution for the covariate (skew=0, kurtotis=3) is the default. Moderate (skew=1, kurtotis=4) and extreme skewness (skew=2, kurtotis=9) are implemented using Ramberg *et al.* (1979) method. The distribution for the covariate will not affect the distribution of the outcome much unless b2 is reasonably large. Note that the covariate, when continuous, is always assumed to be standardised (mean=0 and sd=1) since interactions are included in the models. Users need to take that into consideration when deciding on b2 and b3.

`bcov` Inform the command that the covariate is binary covariate instead of continuous (the default).

`bcb(#)` For binary covariate only, probability (*real*) that xcovar=1. The default is 0.5.

`slcov` Inform the command that the covariate is higher-level (e.g., study-level: recruitment setting) rather than patient-level(the default).

Random effects

`tsq0(#)` Random effect between higher level variance for the intercept. The default value is 0, which assumes homogeneity and no random effects. Heterogeneity for this model factor (intercept) is calculated using tsq0 and errsd. For example $I^2 = 100 * tausq0/(tausq0 + errsd^2)$ and $H^2 = 1/(1 - tausq0/(tausq0 + errsd^2))$. Solving for tausq0 we obtain $tausq0 = (I^2 * errsd^2)/(100 - I^2)$ and $tausq0 = H^2 * errsd^2 - errsd^2$. Although **ipdpower** does not allow I^2 or H^2 inputs for the random effects components, they can be easily calculated using these formulas. Additionally users can use the `di` option to obtain the hypothesised heterogeneity levels for the inputted within- and between- variance parameters (say in a small trial simulation, if unsure of calculations).

`tsq1(#)` Random effect between higher level variance for the exposure. The default value is 0, which assumes homogeneity and no random effects. Heterogeneity for this model factor (exposure) would be calculated using *tsq1* and *errsd*. See `tsq0(#)` for calculation details.

`tsq2(#)` Random effect between higher level variance for the covariate. The default value is 0, which assumes homogeneity and no random effects. Heterogeneity for this model factor (covariate) would be calculated using *tsq2* and *errsd*. See `tsq0(#)` for calculation details.

`tsq3(#)` Random effect between higher level variance for the exposure*covariate interaction term. The default value is 0, which assumes homogeneity and no random effects. Heterogeneity for this model factor (interaction) would be calculated using *tsq3* and *errsd*. See `tsq0(#)` for calculation details.

`dtp0(#)` Distribution for intercept random effect: 0=normal; 1=moderate skew; 2=extreme skew. Normal distribution (skew=0, kurtotis=3) is the default. Moderate (skew=1, kurtotis=4) and extreme skewness (skew=2, kurtotis=9) are implemented using the Ramberg *et al.* (1979) method. In the standard two-stage meta-analysis setting, a non-normal distribution for the random effects has been found to have a small effect on power and coverage - even when the distribution is quite extreme.

`dtp1(#)` Distribution for exposure random effect: 0=normal (default, skew=0, kurtosis=3); 1=moderate skew (skew=1, kurtosis=4); 2=extreme skew (skew=2, kurtosis=9).

`dtp2(#)` Distribution for covariate random effect: 0=normal (default, skew=0, kurtosis=3); 1=moderate skew (skew=1, kurtosis=4); 2=extreme skew (skew=2, kurtosis=9).

`dtp3(#)` Distribution for exposure*covariate interaction random effect: 0=normal (default, skew=0, kurtosis=3); 1=moderate skew (skew=1, kurtosis=4); 2=extreme skew (skew=2, kurtosis=9).

`covmat(`*name*`)` Covariance matrix for normally distributed random effects. Alternative random effects definition to allow modelling of relationships between the random effects components. The matrix needs to be a 4x4 symmetrical non-negative matrix, with the diagonal elements corresponding to the random effects variances for intercept (*name*[1,1]), exposure (*name*[2,2]), covariate (*name*[3,3]) and interaction (*name*[4,4]). Non-normal random effects cannot be modelled using this approach.

Missing data

`missp(#)` Probability (*real*) that outcome is missing, to allow for missing data mechanisms. If option `mar(#)` is defined, data are assumed to be missing under a missing at random (MAR) mechanism. If option `mnar(#)` is defined, data are assumed to be missing under a missing not at random (MNAR) mechanism. If neither `mar(#)` or `mnar(#)` are provided along with `missp(#)`, data are assumed to be missing under a missing completely at random (MCAR) mechanism.

`mar(#)` Odds ratio (*real*) that defines a missing at random (MAR) mechanism. The relationship between the covariate and missingness in the outcome is defined by z=ln(`mar(#)`)*xcovar (i.e a logistic regression model), and the missing data are selected (i.e., set to missing) from the z=1 sub-sample. So a value of 1 implies the mechanism is MCAR, a value above 1 implies that the outcome is more likely to be missing for larger values of the covariate and a value below 1 implies that the outcome is more likely to be missing for smaller values of the covariate.

`mnar(#)` Odds ratio (*real*) that defines a missing not at random (MNAR) mechanism. The relationship between the outcome and missingness in the outcome is defined by z=ln(`mnar(#)`)*y (i.e a logistic regression model), and the missing data are selected (i.e., set to missing) from the z=1 sub-sample. So a value of 1 implies the mechanism is MCAR, a value above 1 implies that the outcome is more likely to be missing for larger values of the outcome and a value below 1 implies that the outcome is more likely to be missing for smaller values of the outcome.

`minum(#)` Number (*integer* > 1) of multiple imputations to be executed. This options informs **ipdpower** that multiple-imputation models will be used and therefore missing data with one of the three available structures (MCAR, MAR, MNAR) need to have been defined. For the imputations, univariate linear, logistic or Poisson regression is used depending on the outcome (see **mi impute**). The imputed datasets are then analysed using **mi estimate** as a prefix, for the seven available models. Note that this process can be time consuming for complex models and binary or count outcomes, while convergence issues are amplified since all imputed datasets must run successfully for **mi estimate** to return results. Therefore 5 imputations are recommended for most models, and 2-3 for non-continuous outcomes and models 5, 6 or 7 (see `model(#)` below).

`mipmm(#)` For a continuous outcome, it informs that missing data will be imputed using a predictive mean matching algorithm (rather than linear regression). The algorithm is computationally more expensive and # defines the number (*integer* ≥ 1) of closest observations (nearest neighbors) to draw from.

Modelling

`model(#)` Specifies the form of the regression model: 1 simple, 2 random effects for intercept, 3-7 various mixed effects options. Model 1 corresponds to a regression with **regress**, **logit** or **poisson**, for continuous, binary and count outcomes respectively. Random effects are not considered at all under these models. Model 2 uses the **xt** family of models, sets the higher level as a panel variable with **xtset** and analyses with **xtreg**, **xtlogit** or **xtpoisson**.Only a random effects component for the intercept is considered under with this set of models. Models 3-7 allow for more advanced modelling options, accounting for various random effect components, but are computationally expensive and do not always converge (using **xtmixed**, **xtmelogit** and **xtmepoisson** which have been renamed in Stata v13 - but we wished to ensure **ipdpower** was compatible with Stata v12). The modelling approaches have been described for **ipdforest** and in the following descriptions we assume the highel level is study (i.e., patients nested within studies), for convenience. Model 3 assumes a fixed common intercept, random exposure effects and fixed effect for the covariate. Model 4 assumes fixed study specific intercepts, random exposure effects and fixed study specific effect for the covariate (which is usually the recommended model for performing individual patient data meta-analysis). Model 5 assumes random study intercepts, random exposure effects and fixed study specific effect for the covariate. Model 6 assumes random study intercepts, random exposure effects and random effects for the covariate. Model 7 assumes random study intercept, random exposure effects, random effects for the covariate and random effects for the interaction. Models 5 and 6 often fail to converge and for model 7 non-convergence is more frequent than convergence.

`clvl(#)` Set confidence level. The default is 95% (alpha level of 5%). See **level**.

`seed(#)` Set initial value of random-number seed, for the simulations. The default is 7. See **set seed**.

`nskip` Add `noskip` option to **xtlogit** or **xtpoisson** to return McFadden's pseudo R^2. This option is only relevant when `model(2)` with `outc(1)` or `outc(2)` is used. Computationally, this approach is more expensive since it additionally fits a full maximum-likelihood model with only a constant for the regression equation be fit (which is used as the base model for the comparison with the final model).

`dnorm` Add `normal` option to **xtpoisson** to assume normally distributed random effects for the intercept. The default is gamma-distributed which is computationally less expensive. Additionally, when the `normal` option is specified, model convergence often fails. This option is only relevant when `model(2)` with `outc(2)`. A skew-normal distribution is similar to gamma and perhaps skew-normal random effects should be considered when modelling count data.

`xnodts` Suppress simulation progress display. If option not specified, a '.' is displayed for each successful model run (i.e., converging) and an 'x' for each unsuccessful iteration.

`di` Display results at the end of the simulation process, including: simulation characteristics, average model fit, average statistics for the outcome, average b0-b3, hypothesised heterogeneity values, power and coverage.

`moreon` Set **more on** (default is off).

3.4. Saved results

ipdpower saves the following scalar results in `r()`:

r(bo)	Average coefficient estimate for the intercept	r(b1)	Average coefficient estimate for the exposure
r(b2)	Average coefficient estimate for the covariate	r(b3)	Average coefficient estimate for the interaction
r(nsim)	Number of simulations	r(nrun)	Number of successful simulations
r(ctime)	Computational time (min)	r(rsq)	Average adjusted or pseudo R^2
r(errsd)	Within-sd (error)	r(consd)	between-sd (intercept)
r(grpsd)	Within-sd (exposure)	r(covsd)	between-sd (covariate)
r(intsd)	Within-sd (interaction)		
r(pow0)	Power to detect b0	r(lpow0)	Power to detect b0, lower CI
r(upow0)	Power to detect b0, upper CI		
r(pow1)	Power to detect b1	r(lpow1)	Power to detect b1, lower CI
r(upow1)	Power to detect b1, upper CI		
r(pow2)	Power to detect b2	r(lpow2)	Power to detect b2, lower CI
r(upow2)	Power to detect b2, upper CI		
r(pow3)	Power to detect b3	r(lpow3)	Power to detect b3, lower CI
r(upow3)	Power to detect b3, upper CI		
r(cov0)	Coverage for b0	r(lcov0)	Coverage for b0, lower CI
r(ucov0)	Coverage for b0, upper CI		
r(cov1)	Coverage for b1	r(lcov1)	Coverage for b1, lower CI
r(ucov1)	Coverage for b1, upper CI		
r(cov2)	Coverage for b2	r(lcov2)	Coverage for b2, lower CI
r(ucov2)	Coverage for b2, upper CI		
r(cov3)	Coverage for b3	r(lcov3)	Coverage for b3, lower CI
r(ucov3)	Coverage for b3, upper CI		

Coefficient estimates, power, coverage, simulations and computational time information is always returned. An R^2 statistic cannot be returned for the more advanced multi-level models with **xtmixed**, **xtmelogit** or **xtmepoisson** or for multiple imputations. With `model(1)`, the average adjusted R^2 is returned from **regress** and the average pseudo R^2 from **logit** or **poisson**. With `model(2)`, the average overall R^2 is returned from **xtreg** and, if the `nskip` option is specified, the average overall McFadden's pseudo R^2 from **xtlogit** or **xtpoisson**. Within-sd is only reported for continuous outcomes and when data are analysed with **xtreg** or **xtmixed**. The between-sd components are only returned if accounted for and estimated by the specified model.

3.5. Example

As an example, we used **ipdpower** to calculate the power in a few designs where the outcome is continuous. First we assumed 20 clusters, a total of 5000 patients, no random effects and the default level for the residual error ϵ ($\sigma = 1$). The generated outcome for the model could be described by $Y_{ij} = 1 + 0.5group_{ij} + 0.3X_{ij} + 0.1group_{ij} * X_{ij} + \epsilon_{ij}$ and we proceeded to analyse using option `model(2)`, i.e., **xtreg** allowing for a random intercept.

```
. ipdpower, sn(1000) ssl(5000) ssh(20) b0(1) b1(0.5) b2(0.3) b3(0.1) model(2)
> di xnodts

model  2
# of converging runs:     1000
```

(*Continued on next page*)

```
computational time(m):     2.1
Characteristics for the outcome (means across clusters)
  mean(grp=0):          0.972
  sd(grp=0):            1.050
  mean(grp=1):          1.488
  sd(grp=1):            1.044
modelled within-study variance
  pooled:               1.000
modelled between-study variance (tau^2)
  exposure:             0.000
  covariate:            0.000
  interaction:          0.000
  intercept:            0.000
modelled heterogeneity, I^2 (range: 0 to 100%)
  exposure:              0.00
  covariate:             0.00
  interaction:           0.00
  intercept:             0.00
modelled heterogeneity, H^2 (range: 1 to +inf)
  exposure:              1.00
  covariate:             1.00
  interaction:           1.00
  intercept:             1.00
mean estimates
  b1 (exposure):        0.500
  b2 (covariate):       0.300
  b3 (interaction):     0.100
  b0 (intercept):       0.999
  R^2(%):               15.83
  within-sd(error):     1.000
  betw-sd(_cons):       0.014
  betw-sd(grp):             .
  betw-sd(covar):           .
  betw-sd(grpXcovar):       .
power to detect effects
  exposure:        100.0(99.6-100.0)
  covariate:       100.0(99.6-100.0)
  interaction:     93.6(91.9-95.0)
  intercept:       100.0(99.6-100.0)
coverage for effects (reported CI includes model beta)
  exposure:        96.3(94.9-97.4)
  covariate:       93.8(92.1-95.2)
  interaction:     93.7(92.0-95.1)
  intercept:       95.9(94.5-97.0)
```

Modelled heterogeneity, modelled within- and between-cluster variances, the number of successful runs and computational time are always provided as outputs. On average, the model explained approximately 15.8% of the variance and performed well in the parameter estimation. Assuming we are interested in capturing the hypothesised interaction effect, the power to detect it was estimated at 93.6%. Coverage for the interaction effect was slightly below nominal at 93.7%.
In the next step we introduce heterogeneity for the exposure.

```
. ipdpower, sn(1000) ssl(5000) ssh(20) b0(1) b1(0.5) b2(0.3) b3(0.1) tsq1(0.5)
> model(2) di xnodts

model  2
# of converging runs:     1000
computational time(m):     4.2
Characteristics for the outcome (means across clusters)
  mean(grp=0):          0.986
  sd(grp=0):            1.052
  mean(grp=1):          1.440
  sd(grp=1):            1.296
```

(*Continued on next page*)

```
modelled within-study variance
  pooled:               1.000
modelled between-study variance (tau^2)
  exposure:             0.500
  covariate:            0.000
  interaction:          0.000
  intercept:            0.000
modelled heterogeneity, I^2 (range: 0 to 100%)
  exposure:             33.33
  covariate:             0.00
  interaction:           0.00
  intercept:             0.00
modelled heterogeneity, H^2 (range: 1 to +inf)
  exposure:              1.50
  covariate:             1.00
  interaction:           1.00
  intercept:             1.00
mean estimates
  b1 (exposure):        0.500
  b2 (covariate):       0.300
  b3 (interaction):     0.101
  b0 (intercept):       0.999
  R^2(%):               13.72
  within-sd(error):     1.056
  betw-sd(_cons):       0.345
  betw-sd(grp):             .
  betw-sd(covar):           .
  betw-sd(grpXcovar):       .
power to detect effects
  exposure:        99.4(98.7-99.8)
  covariate:       100.0(99.6-100.0)
  interaction:     92.6(90.8-94.1)
  intercept:       100.0(99.6-100.0)
coverage for effects (reported CI includes model beta)
  exposure:        25.9(23.2-28.7)
  covariate:       95.5(94.0-96.7)
  interaction:     95.5(94.0-96.7)
  intercept:       100.0(99.6-100.0)
```

The modelled heterogeneity was moderate, with $I^2 = 33.3\%$. On average, the model explained less of the variance in the outcome, approximately 13.7%. The model betas and the residual error (within-sd) were estimated accurately on average but the model assumes a random intercept and underestimates the hypothesised heterogeneity (since the selected modelling approach assumes a random intercept, when a random exposure effect exists in the simulated data). The power to detect the interaction was slightly lower, estimated at 92.6% but coverage was excellent at 95.5%. However, notice the very low coverage for the exposure which was due to the introduced heterogeneity for that factor.
Next, we change the distribution for the errors from normal to extreme skew-normal.

```
. ipdpower, sn(1000) ssl(5000) ssh(20) b0(1) b1(0.5) b2(0.3) b3(0.1) tsq1(0.5)
> model(2) derr(2) di xnodts

model  2
# of converging runs:     1000
computational time(m):     6.3
Characteristics for the outcome (means across clusters)
  mean(grp=0):          0.984
  sd(grp=0):            1.017
  mean(grp=1):          1.518
  sd(grp=1):            1.338
modelled within-study variance
  pooled:               1.000
```

(*Continued on next page*)

```
modelled between-study variance (tau^2)
  exposure:             0.500
  covariate:            0.000
  interaction:          0.000
  intercept:            0.000
modelled heterogeneity, I^2 (range: 0 to 100%)
  exposure:             33.33
  covariate:             0.00
  interaction:           0.00
  intercept:             0.00
modelled heterogeneity, H^2 (range: 1 to +inf)
  exposure:              1.50
  covariate:             1.00
  interaction:           1.00
  intercept:             1.00
mean estimates
  b1 (exposure):        0.503
  b2 (covariate):       0.300
  b3 (interaction):     0.100
  b0 (intercept):       1.002
  R^2(%):               13.72
  within-sd(error):     1.056
  betw-sd(_cons):       0.346
  betw-sd(grp):             .
  betw-sd(covar):           .
  betw-sd(grpXcovar):       .
power to detect effects
  exposure:        99.4(98.7-99.8)
  covariate:       100.0(99.6-100.0)
  interaction:     91.4(89.5-93.1)
  intercept:       100.0(99.6-100.0)
coverage for effects (reported CI includes model beta)
  exposure:        26.2(23.5-29.0)
  covariate:       95.3(93.8-96.5)
  interaction:     92.9(91.1-94.4)
  intercept:       99.9(99.4-100.0)
```

Although the non-normal errors did not appear to affect the model fit, power and coverage were slightly affected. The power to detect the interaction was 91.4% and coverage dropped to 92.9%. Coverage was still very problematic for the exposure.

Next, we proceeded to analyse with a more appropriate model (`model(3)`), one that accounts for the exposure heterogeneity.

```
. ipdpower, sn(1000) ssl(5000) ssh(20) b0(1) b1(0.5) b2(0.3) b3(0.1) tsq1(0.5)
> model(3) derr(2) di xnodts

model  3
# of converging runs:     1000
computational time(m):    14.0
Characteristics for the outcome (means across clusters)
  mean(grp=0):          0.984
  sd(grp=0):            1.017
  mean(grp=1):          1.518
  sd(grp=1):            1.338
modelled within-study variance
  pooled:               1.000
modelled between-study variance (tau^2)
  exposure:             0.500
  covariate:            0.000
  interaction:          0.000
  intercept:            0.000
modelled heterogeneity, I^2 (range: 0 to 100%)
  exposure:             33.33
  covariate:             0.00
  interaction:           0.00
```

(*Continued on next page*)

```
  intercept:             0.00
modelled heterogeneity, H^2 (range: 1 to +inf)
  exposure:              1.50
  covariate:             1.00
  interaction:           1.00
  intercept:             1.00
mean estimates
  b1 (exposure):        0.509
  b2 (covariate):       0.300
  b3 (interaction):     0.100
  b0 (intercept):       0.999
  R^2(%):                   .
  within-sd(error):     0.999
  betw-sd(_cons):           .
  betw-sd(grp):         0.676
  betw-sd(covar):           .
  betw-sd(grpXcovar):       .
power to detect effects
  exposure:        88.7(86.6-90.6)
  covariate:       100.0(99.6-100.0)
  interaction:     93.4(91.7-94.9)
  intercept:       100.0(99.6-100.0)
coverage for effects (reported CI includes model beta)
  exposure:        93.8(92.1-95.2)
  covariate:       95.0(93.5-96.3)
  interaction:     94.3(92.7-95.7)
  intercept:       94.7(93.1-96.0)
```

The model performed well and slightly under-estimated the hypothesised heterogeneity ($\sqrt{0.5} \approx 0.707$). The power to detect the interaction was 93.4% and coverage was 94.3%. As expected, coverage for the exposure was not an issue for this model, since it is a much more accurate reflection of the hypothesised data structure.

Finally, we analysed the data with the recommended model for IPD meta-analyses which is computationally more expensive (`model(4)`). Under this model, fixed cluster specific effects for the covariate and fixed cluster specific intercepts were assumed (i.e., different parameter estimates for each cluster) and therefore covariate and intercept information could not be reported.

```
. ipdpower, sn(1000) ssl(5000) ssh(20) b0(1) b1(0.5) b2(0.3) b3(0.1) tsq1(0.5)
> model(4) derr(2) di xnodts

model  4
# of converging runs:     1000
computational time(m):    46.7
Characteristics for the outcome (means across clusters)
  mean(grp=0):          0.984
  sd(grp=0):            1.017
  mean(grp=1):          1.518
  sd(grp=1):            1.338
modelled within-study variance
  pooled:               1.000
modelled between-study variance (tau^2)
  exposure:             0.500
  covariate:            0.000
  interaction:          0.000
  intercept:            0.000
modelled heterogeneity, I^2 (range: 0 to 100%)
  exposure:             33.33
  covariate:             0.00
  interaction:           0.00
  intercept:             0.00
modelled heterogeneity, H^2 (range: 1 to +inf)
  exposure:              1.50
  covariate:             1.00
```

(*Continued on next page*)

```
  interaction:           1.00
  intercept:             1.00
mean estimates
  b1 (exposure):        0.508
  b2 (covariate):           .
  b3 (interaction):     0.100
  b0 (intercept):           .
  R^2(%):                   .
  within-sd(error):     0.996
  betw-sd(_cons):           .
  betw-sd(grp):         0.666
  betw-sd(covar):           .
  betw-sd(grpXcovar):       .
power to detect effects
  exposure:        88.6(86.5-90.5)
  covariate:         .(  .-  .)
  interaction:     93.2(91.5-94.7)
  intercept:         .(  .-  .)
coverage for effects (reported CI includes model beta)
  exposure:        93.0(91.2-94.5)
  covariate:         .(  .-  .)
  interaction:     93.8(92.1-95.2)
  intercept:         .(  .-  .)
```

The model did not appear to perform better than `model(3)`. The average heterogeneity estimate was similar and so were power and coverage for the interaction at 93.2% and 93.8% respectively. However, a more thorough investigation would be required to make a convincing recommendation on model choice.

4. Discussion

We aimed to describe **ipdpower**, a new simulation-based power calculation command. The command offers a plethora of modelling choices for researchers and its novelty lies with non-normal distribution options and the various random effects assumptions that can be implemented. However, **ipdpower** is flexible and, at its simplest, the higher-level inputs can be ignored to produce power calculation of a one-level model. In addition, if a parameter is not needed (say the covariate or the interaction) the user can just set the respective coefficient to zero when defining the model structure.

Usually researchers wish to input the desired power level and estimate the number of patients and second level units required. Such an approach would be more complicated to implement in **ipdpower** since the parameters of interest are two (number of patients and number of higher-level units) and the combinations that would provide the desired power are many, unless one of them is fixed. Nevertheless, users with a little programming experience should call the command from a binary search algorithm (since power is a monotonic function of sample size), that dichotomises the search area to arrive to the solution quickly, in order to calculate the sample sizes needed to have a desired power level (Cormen, Leiserson, Rivest, Stein *et al.* 2001). Either the average number of patients in a cluster or the number of clusters should be fixed in this approach. Similarly, researchers interested in generating datasets with the specified characteristics, rather than calculating power, can easily obtain them by taking advantage of the fact that **ipdpower** keeps the last simulated dataset in memory. Calling **ipdpower** from within a loop, with `sim(1)` and a different seed number for each iteration, and appending the generated datasets (possibly adding an identifier for each iteration) will produce a sample of datasets.

It might appear confusing that the command accepts variance inputs for the between-cluster heterogeneity and returns standard deviation estimates. In the meta-analysis literature, heterogeneity is usually defined using the variance, but, on the other hand, Stata models typically return standard deviations. We attempted to satisfy both practices.

When modelling continuous predictors (e.g., covariate or exposure), **ipdpower** generates them as standardised (mean=0, sd=1) to avoid potential complications due to the interaction term. However, in most cases users will wish to hypothesise effects for non-standardised variables plus it might not be straightforward to define mean levels for outcomes through `b0(#)` (e.g., $Pr(Y = 1|group = 0)$ for a binary outcome). To help researchers in setting up the correct design we have provided an accompanying Microsoft Excel file with examples across all possible exposure-covariate scenarios, with or without interaction effects (from http://www.statanalysis.co.uk/files/defining_betas.xlsx). The command also returns aggregated simulation statistics with option `di`, allowing users to double-check the hypothesised coefficients and adjust if necessary.

When many random effects are modelled, assuming that all correlations between them are zero might not always be a realistic assumption. From a practical point of view, these correlations might have very small effects on performance while their hypothesised values for a particular study could be anyone's guess. Nevertheless, we have allowed relationships between normally distributed effects to be specified using a covariance matrix (option `covmat(name)`), and modelled with the **drawnorm** command. Such an approach was not possible when modelling non-normal random effects, however, and all correlations between non-normal effects are assumed to be zero.

ipdpower uses some of the mixed effects modelling commands that were renamed in Stata v13; `xtmixed` to `mixed`, `xtmelogit` to `meqrlogit` and `xtmepoisson` to `meqrpoisson`. However, we wished to ensure Stata v12 users would have access to the command and our choice does not affect users of later versions.

Future work could involve using **ipdpower** as a platform to inform on the effectiveness of multiple-imputation mechanisms, especially in Primary Care Databases under MNAR assumptions. In addition, although one-stage IPD meta-analysis is considered to be more robust from a theoretical perspective (Mathew and Nordstrom 2010), the simplicity of a two-stage IPD analysis and circumstantial evidence where results were very close with the two approaches make the latter appealing to researchers. However, the asymptotic nature of the methods implies that a theoretical comparison does not provide a complete picture and a simulation study is also needed to compare the two approaches, especially for small numbers of meta-analyses studies and in the presence of heterogeneity. Until such comparisons are made we must warn researchers that heterogeneity estimates in two-stage aggregate meta-analyses can be very inaccurate and more often than not seem to fail to account for existing heterogeneity (Kontopantelis and Reeves 2012; Kontopantelis *et al.* 2013), and these methodological problems are likely to be an issue in two-stage IPD meta-analyses as well.

Acknowledgments

This study was funded by the National Institute for Health Research (NIHR) School for Primary Care Research (SPCR), under the title 'An analytical framework for increasing the efficiency and validity of research using primary care databases' (Project no. 211). This

paper presents independent research funded by the National Institute for Health Research (NIHR). The views expressed are those of the authors and not necessarily those of the NHS, the National Institute for Health Research or the Department of Health.

Contribution

EK designed and developed the command and wrote the manuscript. DS, RP and DR critically commented on both the manuscript and the functionality of the command.

References

Arnold BF, Hogan DR, Colford Jr JM, Hubbard AE (2011). "Simulation Methods to Estimate Design Power: An Overview for Applied Research." *BMC Med Res Methodol*, **11**, 94. doi:10.1186/1471-2288-11-94. URL http://dx.doi.org/10.1186/1471-2288-11-94.

Brockwell SE, Gordon IR (2001). "A Comparison of Statistical Methods for Meta-Analysis." *Statistics in medicine*, **20**(6), 825–840. URL http://onlinelibrary.wiley.com/doi/10.1002/sim.650/full.

Cormen TH, Leiserson CE, Rivest RL, Stein C, *et al.* (2001). *Introduction to Algorithms*, volume 2. MIT press Cambridge.

Feiveson A (2002). "Power by Simulation." *Stata Journal*, **2**(2), 107–124.

Higgins J, Thompson SG (2002). "Quantifying Heterogeneity in a Meta-Analysis." *Statistics in medicine*, **21**(11), 1539–1558. URL http://onlinelibrary.wiley.com/doi/10.1002/sim.1186/full.

Higgins JP, Whitehead A, Turner RM, Omar RZ, Thompson SG (2001). "Meta-analysis of Continuous Outcome Data from Individual Patients." *Stat Med*, **20**(15), 2219–2241. doi:10.1002/sim.918. URL http://dx.doi.org/10.1002/sim.918.

Hunter JE, Schmidt FL (2000). "Fixed Effects vs. Random Effects Meta-Analysis Models: Implications for Cumulative Research Knowledge." *International Journal of Selection and Assessment*, **8**(4), 275–292. ISSN 1468-2389. doi:10.1111/1468-2389.00156. URL http://dx.doi.org/10.1111/1468-2389.00156.

Kontopantelis E, Reeves D (2012). "Performance of Statistical Methods for Meta-Analysis when True Study Effects are non-Normally Distributed: A Simulation Study." *Statistical methods in medical research*, **21**(4), 409–426. URL http://smm.sagepub.com/content/21/4/409.short.

Kontopantelis E, Reeves D (2013). "A Short Guide and a Forest Plot Command (ipdforest) for One-Stage Meta-Analysis." *Stata Journal*, **13**(3), 574–587.

Kontopantelis E, Springate DA, Reeves D (2013). "A Re-Analysis of the Cochrane Library Data: The Dangers of Unobserved Heterogeneity in Meta-Analyses." *PloS one*, **8**(7), e69930.

Kovalchik SA (2013). "Aggregate-Data Estimation of an Individual Patient Data Linear Random Effects Meta-Analysis with a Patient Covariate-Treatment Interaction Term." *Biostatistics*, **14**(2), 273–283. URL http://biostatistics.oxfordjournals.org/content/14/2/273.short.

Kovalchik SA, Cumberland WG (2012). "Using Aggregate Data to Estimate the Standard Error of a Treatment–Covariate Interaction in an Individual Patient Data Meta-Analysis." *Biometrical Journal*, **54**(3), 370–384. URL http://onlinelibrary.wiley.com/doi/10.1002/bimj.201100167/full.

Mathew T, Nordstrom K (2010). "Comparison of One-Step and Two-Step Meta-Analysis Models using Individual Patient Data." *Biom J*, **52**(2), 271–287. doi:10.1002/bimj.200900143. URL http://dx.doi.org/10.1002/bimj.200900143.

McClelland GH, Judd CM (1993). "Statistical Difficulties of Detecting Interactions and Moderator Effects." *Psychol Bull*, **114**(2), 376–390.

Mittlbock M, Heinzl H (2006). "A Simulation Study Comparing Properties of Heterogeneity Measures in Meta-Analyses." *Statistics in Medicine*, **25**(24), 4321–4333.

Ramberg JS, Dudewicz EJ, Tadikamalla PR, Mykytka EF (1979). "A Probability Distribution and its Uses in Fitting Data." *Technometrics*, **21**(2), 201–214. URL http://www.tandfonline.com/doi/abs/10.1080/00401706.1979.10489750.

R Development Core Team (2013). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

StataCorp (2011). *Stata Statistical Software: Release 12*. StataCorp LP, College Station, TX.

Turner RM, Omar RZ, Yang M, Goldstein H, Thompson SG (2000). "A Multilevel Model Framework for Meta-Analysis of Clinical Trials with Binary Outcomes." *Stat Med*, **19**(24), 3417–3432.

Whitehead A (2002). *Meta-Analysis of Controlled Clinical Trials*. Statistics in Practice. John Wiley & Sons Inc.

Affiliation:

Evangelos Kontopantelis
Senior Research Fellow
Centre for Health Informatics & Centre for Primary Care
NIHR School for Primary Care Research
Institute of Population Health
University of Manchester
E-mail: e.kontopantelis@manchester.ac.uk
URL: http://www.population-health.manchester.ac.uk/staff/EvanKontopantelis

David A Springate
Research Fellow
Institute of Population Health
University of Manchester
E-mail: david.springate@manchester.ac.uk
URL: http://www.population-health.manchester.ac.uk/staff/daspringate

Rosa Parisi
Research Associate
Manchester Pharmacy School
University of Manchester
E-mail: rosa.parisi@manchester.ac.uk
URL: http://www.pharmacy.manchester.ac.uk/staff/rosaparisi

David Reeves
Reader
Institute of Population Health
University of Manchester
E-mail: david.reeves@manchester.ac.uk
URL: http://www.population-health.manchester.ac.uk/staff/76561

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume VV, Issue II *Submitted:* yyyy-mm-dd
MMMMMM YYYY *Accepted:* yyyy-mm-dd