A Monte Carlo Analysis of Alternative Meta-Analysis Estimators in the Presence of Publication Bias

W. Robert Reed, Raymond J.G.M. Florax, and Jacques Poot

Abstract
This study uses Monte Carlo analysis to investigate the performances of five different meta-analysis (MA) estimators: the Fixed Effects (FE) estimator, the Weighted Least Squares (WLS) estimator, the Random Effects (RE) estimator, the Precision Effect Test (PET) estimator, and the Precision Effect Estimate with Standard Errors (PEESE) estimator. The authors consider two types of publication bias: publication bias directed against statistically insignificant estimates, and publication bias directed against wrong-signed estimates. Finally, the authors consider three cases concerning the distribution of the "true effect": the Fixed Effects case, where there is only one estimate per study, and all studies have the same true effect; the Random Effects case, where there is only one estimate per study, and there is heterogeneity in true effects across studies; and the Panel Random Effects case, where studies have multiple estimates, and true effects are random both across and within studies. The simulations produce a number of findings that challenge results from previous research.

(Published in Special Issue Meta-Analysis in Theory and Practice)

Authors
W. Robert Reed, ✉ Department of Economics and Finance, University of Canterbury, Christchurch, New Zealand, bob.reed@canterbury.ac.nz
Raymond J.G.M. Florax, Department of Agricultural Economics, Purdue University, USA & Department of Spatial Economics, VU University Amsterdam & Tinbergen Institute, The Netherlands
Jacques Poot, National Institute of Demographic and Economic Analysis & Department of Economics, University of Waikato, New Zealand

I. INTRODUCTION

A meta-analysis (MA) is a systematic evaluation of a body of research that measures some "effect," such as the effect of minimum wages on unemployment, or the effect of an increase in electricity prices on residential demand. It is no exaggeration to say that most areas of empirical study in economics are characterized by disparate, and often conflicting, estimates. Meta-analysis is an attempt to summarize and "make sense" of these disparate findings. Meta-analyses have a long tradition in the medical sciences, and are increasingly popular in economics.

A major concern in the MA literature is the problem of "publication bias." Publication bias occurs when the sample of research that gets "published" does not represent the population of studies that measure the effect.[1] This can happen when journals reject studies that report statistically insignificant estimates, or that are "wrong-signed." It can also happen when researchers anticipate rejection by journals due to undesirable results, and choose either to not submit their work to journals (the "file drawer" effect); or submit only those regressions that are most supportive of their hypothesis.[2]

While small sample size/statistical insignificance has received the preponderance of attention, both types of publication bias have been identified in the literature as important. For example, three of the four cases analysed in-depth by Stanley and Doucouliagos (2012) – namely: Dalhuisen et al. (2003); Bellavance et al. (2009) and Doucouliagos and Stanley (2009) – focus on publication bias due to (perceived) wrong signs.

Despite a voluminous literature, there are relatively few studies that examine the performance of meta-analysis procedures designed to address publication bias. Stanley (2008) compares the performance of meta-analysis estimators on the dimensions of power,

---

[1] Following conventional usage, we use "publication bias" to refer both to (i) the sample selection that occurs when research is non-randomly selected into publication outlets, and (ii) the bias in estimated effects that arises from that selection process.

[2] The latter kind of publication bias can be reduced by journals insisting on extensive robustness checks and meta-analysts including all such robustness checks in their analysis.

size, and mean squared error. His analysis produces a relatively sanguine evaluation of the ability of meta-analysis estimators to reliably detect, and estimate, variable effects in the presence of publication bias: "Meta-regression methods are found to be robust against publication selection. Even if a literature is dominated by large and unknown misspecification biases, precision-effect testing and joint precision-effect and meta-significance testing can provide viable strategies for detecting genuine empirical effects" (Stanley, 2008, p. 103).

Moreno et. al (2009) compares a large number of meta-analysis estimators on the dimensions of bias, coverage rates, mean squared error, and variance. They also come to an overall positive evaluation, at least for a subset of the estimators: "In this paper we have compared some novel and existing methods for adjusting for publication bias through an extensive simulation study. Results are encouraging, with several of the regression methods displaying good performance profiles" (Moreno et al. 2009, p. 12).

Two recent papers by Stanley and Doucouliagos (2014a, 2014b) follow on earlier work by Koetse et al. (2010) and promote another meta-analysis estimator that they call the "Weighted Least Squares" (WLS) meta-analysis estimator.[3] Stanley and Doucouliagos argue that this estimator outperforms both the conventional fixed effects estimator and the random effects estimator in the presence of publication bias.

While all these studies make important contributions, they leave significant gaps in their coverage. For example, it is common to only consider two scenarios, "no effect" and "effect", and to ignore the interaction of effect size and publication bias. Previous studies that focus on publication bias assume each study only produces one estimate. They do not consider scenarios where studies produce multiple estimates of an effect, a common feature

---

[3] See Section 3.3 in Koetse et al. (2010), specifically footnote 6.

of economic studies. Finally, there is typically little effort made to ensure that the simulated samples "look like" the kinds of samples used in actual meta-analysis studies.

Our research simulates populations of studies, and subjects those studies to publication bias. It will then compare a variety of estimators for measuring the "true" effect, or the mean "true" effect, when the meta-analyst's sample is subject to publication bias. The analysis will focus on three cases. The "Fixed Effects" case arises when all studies have the same effect in the population, but sampling variation causes different studies to produce different estimates of that effect. In this case, the meta-analyst attempts to identify the single "true" effect. Given the common heterogeneity across studies in economics, this case is the least interesting but it provides a useful benchmark. The "Random Effects" case occurs when studies estimate different population effects, but with a common expected value. In this case, the meta-analyst attempts to identify this expected value, i.e. the average of the individual "true" effects.

These first two cases examine scenarios where only one estimated effect is produced by/taken from each study. The third case addresses the situation where studies contain multiple estimates. In this case, each study has a different "true" effect that varies both within and across studies due to specification and other issues. This is arguably the most common case in economics.

The results presented in this study are based on the particular specifications of the data-generation process used in our Monte Carlo simulations. However, they are sufficient to support the following conclusions. First, MA estimators that correct for publication bias sometimes perform worse than those that do not. Second, MA estimators perform poorly when there is substantial heterogeneity in true effects, and publication bias affects the most precise estimates. Third, while random effects estimators are often more biased than other MA estimators, they sometimes are more efficient. And finally, hypothesis testing about the

3

mean true effect is generally unreliable for all MA estimators. While previous research has identified specific experimental settings which demonstrate the superiority of one or more MA estimators, our results show that these findings do not generalize to all settings. There are variations in the experimental design of existing simulation studies whose impact on the small sample characteristics of meta-analysis estimators are unknown. A general conclusion therefore remains elusive for now.

II. FIXED EFFECTS

IIA. FIXED EFFECTS: DESIGN OF MONTE CARLO EXPERIMENTS

SIMULATING THE STUDIES. All of our Monte Carlo experiments follow a two-stage procedure. The first stage consists of generating a population of "studies." In the Fixed Effect (FE) case, we construct a population of M studies. Each meta-analysis study has one estimated effect based on a sample of T observations (later we will generalise this to multiple estimates per study). The DGP generating the simulated observations for a given study i is given by:

(1) $y_{it} = 1 + \alpha\, x_{it} + e_{it}$, $t = 1,2,\ldots,T,$

where the "true" effect a is a parameter that we vary across Monte Carlo experiments.

At the level of the individual study, these observations are used to estimate the following specification:

(2) $y_{it} = \beta_{i0} + \beta_{i1} \cdot x_{it} + error_{it}$, $t = 1,2,\ldots,T.$

The estimated effect for study i is given by the OLS estimate of β_{i1} in Equation (2). The assumptions regarding e_{it} in Equation (1) imply that OLS will in each study yield a best linear unbiased estimate of α.

While the true effect is the same for all studies, and the distribution of the x's are the same ($x\sim NID(0,1)$), the studies differ in the variances of their respective error terms. This causes the OLS estimates across different studies to have different standard errors.

4

A critical component in our Monte Carlo simulations is the specification of the error term in Equation (2). We choose parameter values that attempt to satisfy three criteria:

1. Produce a realistic range of t-values for the estimated effects in the population of studies

2. Produce realistic-looking funnel graphs

3. Cause the percent of studies that "disappear" through publication bias to range between 0 and 80 percent.

In the FE case, the error term for study i is simulated by

(3.A) $e_{it} = \lambda_i \cdot NID(0,1)$, $t = 1,2\ldots,T$, where

(3.B) $\lambda_i = 0.2 + 30*UID(0,1)$, $i = 1,2,\ldots,M$;

where $UID(0,1)$ indicates that this component of the error term is uniformly and independently distributed on support [0,1]. This formulation allows a floor to be set for the minimum value of the variance of the error term (0.04). This prevents small error variances from generating unreasonably large t-statistics for the estimated effects.

For each study i, we record the estimated value of β_{i1} and its standard error. In the Fixed Effects case, we repeat the above process M times, so that there is a sample of M studies, each with its own estimated effect and standard error (but common true effect). This sample constitutes the population of all estimated effects. In the actual simulations, we will set $T = 100$ and $M = 300$.

PUBLICATION BIAS. Two types of publication bias are modelled. The first type assumes that the publication process discriminates in favour of studies that have statistically significant estimates, indicated by t-statistics with absolute values greater than or equal to 2. Studies with insignificant estimates can still get "published," but only with probability equal to 0.10. The second type of publication bias discriminates against studies that have estimated effects of the "wrong sign." Without loss of generality, we assume that economic theory posits that the correct sign of β_1 should be positive (as in value of life), so that negative

estimated values have a much harder time getting published. Their probability of getting published is again set equal to 0.10.

SIX DIFFERENT ESTIMATORS OF EFFECT SIZE. The second stage of our analysis simulates the meta-analysis procedure. Given a set of studies, meta-analysis aims to measure the true effect of x on y. There are a variety of ways to do this. Our Monte Carlo analysis compares the performances of six different estimators: "Unadjusted" average (= OLS), and five different MA estimators, two of which are specifically designed to address publication bias.

The "Unadjusted" estimator: This estimator calculates a simple average of the reported estimated effects from the publication-biased sample of studies available to the meta-analyst. Let $\hat{\beta}_i$ be the estimated effect from study i. The Unadjusted estimator of the true effect is the OLS estimate of α in the equation below:

(4) $\qquad \hat{\beta}_{i1} = \alpha + v_i, \ i = 1,2,\ldots,M,$

where M is the number of published studies available to the meta-analyst. Clearly, the unadjusted estimator simply calculates the arithmetic mean of estimated effects across studies.

The "Fixed Effects" (FE) estimator. The FE estimator weights all the observations by the inverse of the estimated standard error of $\hat{\beta}_i$, SE_i. It assumes that any variation in the estimated effects across studies is due solely to sampling error. The FE estimator of the true effect is the weighted least squares estimate of α in Equation (4):

(5) $\qquad \dfrac{\hat{\beta}_{i1}}{SE_i} = \alpha \cdot \left(\dfrac{1}{SE_i}\right) + \dfrac{v_i}{SE_i}, \ i = 1,2,\ldots,M;$

except that the residuals are standardized to produce a sample variance of 1.

The "Weighted Least Squares" (WLS) estimator. The WLS estimator is identical to the FE estimator except that the residuals remain unstandardized. Note that the FE and WLS

estimators produce identical estimates of α, but the associated estimates have different standard errors.

The "Random Effects" (RE) estimator. The "RE" estimator allows for heterogeneity in effects above and beyond what would be expected from sampling variation. It assumes that differences in estimated effects across studies are due to (i) sampling variation, and (ii) genuine differences in the underlying effects. This second component is represented by τ, which is the standard deviation of underlying effects across studies. If the two error components are independent, then the s.e.$(\hat{\beta}_i) = \sqrt{(SE_i)^2 + \tau^2} = \omega_i$. The RE estimator of the mean true effect (the mean of the distribution of random true effects), is given by weighted least squares estimation of α in Equation (4), with weights $= \omega_i$:

(6) $\qquad \dfrac{\hat{\beta}_{i1}}{\omega_i} = \alpha \cdot \left(\dfrac{1}{\omega_i}\right) + \dfrac{v_i}{\omega_i}, \;\; i = 1,2,\ldots,M.$

The Precision Effect Test (PET) estimator: The first of two MA estimators designed to address publication bias is the "PET" estimator. The name of this estimator derives from the fact that it is associated with a particular test for publication bias known as the Precision Effect Test (PET). The PET adds the ith study's estimated standard error of the estimated effect, (SE_i) to Equation (4):

(7) $\qquad \hat{\beta}_{i1} = \alpha + \alpha_1 \cdot SE_i + v_i, \; i = 1,2,\ldots,M.$

WLS estimation of α in Equation (7) provides an estimate of the true effect of x on y, α, adjusting for publication bias α_1:

(8) $\qquad \dfrac{\hat{\beta}_{i1}}{SE_i} = \alpha \cdot \left(\dfrac{1}{SE_i}\right) + \alpha_1 + \dfrac{v_i}{SE_i}, \;\; i = 1,2,\ldots,M).$

The Precision Effect Estimator with Standard Errors (PEESE) estimator: The second MA estimator designed to address publication bias is the "PEESE" estimator. This estimator is identical to the PET estimator, except that it replaces SE_i with $(SE_i)^2$ in Equation (7). This yields the following weighted least squares specification,

(9) $\quad \dfrac{\hat{\beta}_{i1}}{SE_i} = \alpha_0 \cdot \left(\dfrac{1}{SE_i}\right) + \alpha_1 \cdot SE_i + \dfrac{v_i}{SE_i}, \quad i = 1,2,\dots,\text{M}.$

Note that there are no constant terms in the specifications of Equations (5), (6) and (9).

Both the PET and PEESE estimators correct for publication bias by adding some form of the effect's standard error to the regression specification. The rationale for this approach is loosely linked to Heckman-type procedures for correcting sample selection bias (Stanley and Doucouliagos, 2012, pp. 117ff.).[4]

PERFORMANCE MEASURES. Our analysis employs three performance measures: (i) Bias, (ii) Mean Squared Error (MSE), and (ii) accuracy of hypothesis testing, as measured by Type I error rates associated with testing the hypothesis: $H_0: true\ value = \alpha$. We simulate 1,000 meta-analysis studies, where each meta-analysis study is generated as discussed above. Further, we do this for 9 different values of true effects, $a = 0.0, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5,$ and $4.0.$[5]

Note that the extent of publication bias is affected by the value of α. When α = 0, there will be more studies with statistically insignificant estimated effects, and more studies with wrong-signed estimated effects, than when α = 4. However, the consequences will differ depending on the type of publication bias. For example, when α = 0, publication bias based on statistical insignificance will not produce biased estimates of α. In contrast, publication bias against wrong-signed estimates will have its greatest impact on estimator bias when α = 0.

IIB. FIXED EFFECTS: AN EXAMPLE META-ANALYSIS SAMPLE (α = 1)

TABLE 1 reports the distribution across studies of (i) estimated effects, (ii) *t*-statistics, and (iii) precisions of the estimated effect for a representative meta-analysis sample, where

---

[4] While it may be semantically more accurate to refer to the PET and PEESE as alternative specifications of the WLS estimator, we will refer to them as estimators for ease of exposition.
[5] The Stata programs for generating the results of Tables 2, 3, 5, 6, 8, and 9 are included in the Appendix to this paper. Additional programs are available from the corresponding author upon request.

representative is defined as average values across 1000 simulations.[6] The sample was

constructed using the design parameters described above, with the true effect set to one (α =

1). The "pre-publication bias" panel of the table reports sample characteristics for the

population of 300 studies potentially available to the meta-analyst. The next two panels in

the table summarize the "published literature" available to the meta-analyst after each type of

publication truncation is applied to the population of studies.

When α = 1, the two types of publication bias cause, respectively, approximately 68

percent and 19 percent of the population studies to be omitted from the meta-analyst's sample

of studies.   The average minimum and maximum values of estimated effects in the

population are [-5.74,7.76].[7] In the publication-biased samples, the average ranges are

[-5.01,7.65] and [-3.10,7.72], respectively. The average estimation bias associated with the

median study in the two population-biased samples is approximately 20%.

The median t-statistic in the population of studies is 0.94. This compares to median t

values of 2.62 and 1.27 in the two publication-biased samples. Note that even when

publication bias discriminates against insignificant results, there are still some studies that

have low t-statistics due to the assumed probabilistic nature of the bias/selection process

(10% are still published). The precision (1/SE) is used in a number of MA procedures to

weight individual observations (see above). Relative weights across different observations

can differ by a factor of 50 or more. The differences in t-statistics and precisions across the

publication-biased samples illustrates that the standard error variable, used in the PET and

PEESE corrections for publication bias, is itself endogenous to publication bias.

The purpose of TABLE 1 is to demonstrate that the parameters chosen for our Monte

Carlo experiments produce samples that approximately "look like" those used in actual meta-

---

[6] Empirical results are based on computer programs using Stata, version 13.1. The associated .do files for all
tables and figures are included in the Appendix.
[7] These are "average" minimum and maximum values because they are the minimum and maximum values
averaged over the 1000 simulated meta-analysis samples.

analyses. While meta-analysis samples certainly differ, the range of estimated effects, *t*-statistics, and precision values in TABLE 1 would not attract attention for being unusual. Since external validity is always an issue with Monte Carlo studies, good practice should ensure that the samples under study are similar to ones that occur in actual empirical work.

FIGURE 1 presents a series of representative funnel graphs for each of the three samples, assuming the same parameter values used in TABLE 1. The solid vertical line represents the true effect ($\alpha = 1$). The top figure shows a funnel graph for the population of 300 studies prior to any publication bias. The figure demonstrates the classic funnel shape, with the distribution of estimates being centred around the true effect, and approximately equal numbers of studies on either side of the true effect for a given SE value.

The next figure shows the same set of studies after they have been subject to publication bias against statistical insignificance. As noted in TABLE 1, publication bias against statistical insignificance has a large impact on the meta-analysis sample when $\alpha = 1$, with approximately 68% of studies, on average, never seeing the light of day. The remaining 32% present a skewed sample of estimated effects. While the most precise estimates are still centred closely around the true effect, less precise estimates are disproportionately distributed to the right of the true effect.

The last funnel graph works with the same population of studies represented in the top figure, but applies a selection criterion that disproportionately favours studies that have the correct/positive sign. Again, the most precise estimates are closely clustered around the true effect, while less precise estimates are disproportionately located to the right of the true effect. One thing that is immediately apparent from these funnel graphs is that any estimation procedure that heavily weights more precise estimates is likely to produce unbiased estimates

of the true effect. While this is often the case, as we show below, it is particularly true for the fixed effects case.[8]

IIC. FIXED EFFECTS: PERFORMANCE TESTS

TABLES 2 and 3 compare the six different estimators across three performance dimensions: (i) Mean Estimate of True Effect, (ii) Mean Squared Error (MSE), and (iii) Type I error rates associated with the hypothesis $H_0: true\ value = \alpha$. We use a significance level of 5 percent, so rejection rates should likewise be equal to 5 percent. TABLE 2 reports the results when publication bias is directed against statistical insignificance. TABLE 3 examines publication bias against wrong-signed estimates. Each of the estimators is studied for a set of true effect values (= α) ranging from 0.0 to 4.0 in half unit steps.

TABLE 2. The top panel reports the average estimated effects for the respective estimators. The first two columns report the value of the true effect (α) and the percent of studies included in a typical meta-analysis study (*N*). The first thing to note is that there is a strong relationship between the size of the true effect and the number of studies that survive publication bias against statistical insignificance. When α = 0, less than 15% of all studies appear in the meta-analyst's sample. As α increases and the mean of the distribution of estimated effects moves away from zero, more and more studies produce significant estimates. When α = 4, approximately three-fourths of all studies survive publication bias and are included in the meta-analyst's sample.

The next column reports results for the Unadjusted estimator. When α = 0 and publication bias discriminates against insignificant estimates, the Unadjusted estimator remains an unbiased estimator of the true effect. This is because sampling error is equally likely to produce significant estimates that are above and below the true effect. However, as α increases, publication bias disproportionately omits studies with estimates below the true

---
[8] This highlights a point previously made by Stanley, Jarrell, and Doucouliagos (2010).

effect since, ceteris paribus, studies with small estimated effects are more likely to have small *t*-values. When α = 1.0, the Unadjusted estimator overestimates the true effect by approximately 57 percent. As α increases, fewer and fewer studies are affected by publication bias. While the table does not show this, further increases in α would cause the publication bias associated with the Unadjusted estimator to eventually disappear.

Continuing with the top panel of TABLE 2, we turn our attention to the performances of the five MA estimators (PET, PEESE, FE, WLS, and RE). We are particularly interested in the first two estimators, which are specifically designed to address publication bias. With respect to estimation bias, both estimators do very well. When α = 1, the mean estimates of the true effect for the PET and PEESE estimators are 0.96 and 1.01, respectively. When α = 2, the estimates are 1.96 and 2.00, respectively. In fact, for every value of α, the PET and PEESE estimators virtually eliminate estimation bias. This success seemingly validates the ability of the PET and PEESE estimators to correct for publication bias.

However, the next three columns demonstrate that the other MA estimators also perform very well, even though they do not explicitly correct for publication bias. The explanation lies in how the study estimates are weighted. In one way or another, all five MA estimators weight by the inverse of the estimated coefficient's standard error. As seen from the funnel graphs in FIGURE 1, any estimator that heavily weights precise estimates is likely to produce estimated effects close to the true effect, at least in the Fixed Effects case. In other words, while the PET and PEESE estimators are successful in eliminating publication bias, their success has little to do with the inclusion of an *SE* term in the specification (see Equations (7) through (9)).

Turning to the middle panel of TABLE 2, which focuses on MSE performance, we see that the Unadjusted estimator, unsurprisingly, performs by far the worst. All five MA estimators are substantially more efficient. Of the two MA estimators designed to correct for

publication bias, the PEESE estimator is most efficient. However, it is, at best, only marginally better than the FE/WLS estimator.[9]

However, and turning now to the bottom panel, we see that the PEESE estimator struggles with reliability in hypothesis testing, as do several of the other MA estimators. Type I error rates associated with the hypothesis $H_0: true\ value\ =\ \alpha$ are often far in excess of their expected value of 5%. Of the five MA estimators, only the FE estimator produces reliable hypothesis test results.

In summary, while all five MA estimators perform better than the Unadjusted estimator, there is little to recommend the PET and PEESE estimators over the FE estimator in these experiments. Indeed, when it comes to hypothesis testing, the FE estimator, which does not specifically address publication bias, has clear advantages, with relatively little, or no cost, in terms of efficiency.

TABLE 3. TABLE 3 continues investigating the Fixed Effects case, where all studies share a common true effect, but it analyses publication bias that is targeted towards wrong-signed estimates. As evidenced by the top panel, the Unadjusted estimator again produces effect estimates that can be substantially biased. In contrast to publication bias against statistical insignificance, the bias is greatest for small values of α. As α increases, studies estimate fewer negative effects, so more studies get "published." When α is very large (= 4), almost all studies gets published (98.0 percent), and the Unadjusted estimator correspondingly has a relatively small estimation bias.

Turning now to the PET and PEESE estimators, we see that their relative performance on the dimensions of estimation bias and MSE depends on whether α = 0 or α > 0. When α = 0, the PET estimator is superior on these dimensions; when α > 0, the PEESE estimator

---

[9] As noted above, the FE and WLS estimators produce identical coefficient estimates, differing only in the standard errors of the coefficients.

dominates.[10]   As before, the other three MA estimators generally also do a good job of eliminating estimation bias; and have MSE performance similar to the PET and PEESE estimators, except when α is small. When it comes to hypothesis testing, the MA estimators that do not correct for publication bias generally do a better job than the PET and PEESE estimators, except, again, when α is small. None of the estimators are reliable for hypothesis testing for small values of α.

III. RANDOM EFFECTS

IIIA. RANDOM EFFECTS: DESIGN OF MONTE CARLO EXPERIMENTS

SIMULATING THE STUDIES. The Monte Carlo experiments for the Random Effects case follow the same two-stage procedure used in the Fixed Effects case. In the first stage, 300 individual studies are simulated. In the second stage, samples are subjected to publication bias, analysed using meta-analysis procedures, and evaluated for estimator performance. The only difference is how study effects are simulated. In the Fixed Effects case, all studies were assumed to have the same true effect, with different estimated effects due solely to sampling error. In the Random Effects case, studies differ in their true effects, although their true effects have the same expected value.

In our simulations, we generate heterogeneity in true effects across studies by letting the true effect be random, with a mean value of α. Thus Equation (1) from the FE case becomes:

(1'.A) $y_{it} = 1 + \alpha_i \cdot x_{it} + e_{it},\ t = 1,2,\ldots,T$, where

(1'.B) $\alpha_i = N(\alpha, \tau^2)$.

Study-specific effects are obtained the same as in the Fixed Effects case, see Equation (2), by estimating the specification $y_{it} = \beta_{i0} + \beta_{i1} \cdot x_{it} + error_{it}$.

---

[10] This is consistent with the recommendations of Stanley and Doucouliagos (2012, cf.page 78f.)

The only other difference with the Fixed Effects case for the Monte Carlo simulations concerns the generation of the error terms in (1'.A). In the Fixed Effects case, we generated the error terms following Equations (3.A) and (3.B), repeated below.

(3.A) $e_{it} = \lambda_i \cdot NID(0,1)$, where

(3.B) $\lambda_i = 0.2 + 30*UID(0,1)$.

In the Random Effects case, Equation (3.B) is changed to

(3'.B) $\lambda_i = 0.5 + 30*UID(0,1)$

Similar to the FE case, we choose parameter values for the DGP that aim to satisfy the three, previously mentioned criteria:

1. Produce a realistic range of t-values for the estimated effects in the population of studies

2. Produce realistic-looking funnel graphs

3. Cause the percent of studies that "disappear" through publication bias to range between 0 and 80 percent.

In addition, we add a fourth criterion:

4. Produce realistic values of "effect heterogeneity"

"Effect heterogeneity" consists of differences in true effects across studies. A measure of effect heterogeneity is I, which provides an estimate of the total percent of variation in estimated effects that is due to factors other than sampling error (Higgins and Thompson, 2002; Higgins et al., 2003). I^2 values of 70-95% are common in economics studies. For example, Stanley and Doucouliagos (2014a, p. 14) report that "among minimum wage elasticities, I^2 is 90% (Doucouliagos and Stanley, 2009); it is 93% among estimates of the value of statistical life (Doucouliagos, Stanley and Giles, 2012) and 97% among the partial correlations of CEO pay and corporate performance (Doucouliagos, Haman and Stanley, 2012)." The RE DGP parameter values are set so generate I^2 values between 70-95%, while simultaneously satisfying the other three criteria.

IIIB.   RANDOM EFFECTS: AN EXAMPLE META-ANALYSIS SAMPLE ($\alpha = 1$)

TABLE 4 does the same thing for the Random Effects case that TABLE 1 did for Fixed Effects, reporting the distribution of (i) estimated effects, (ii) t-statistics, and (iii) precisions of estimated effects for a representative meta-analysis sample.   In addition, the table adds information about the distribution of I^2, a measure of effect hetereogeneity.   Whereas TABLE 1 reported the average of simulated meta-analysis samples when the true effect was 1, TABLE 4 reports average characteristics when *the mean* of the distribution of true effects is 1.

The sample characteristics do not look dramatically different.   The two types of publication biases reduce the population of 300 studies to samples of approximately 100 and 225 studies, respectively.   These are close to the values reported in TABLE 1.   However, the size of the bias is somewhat larger, with publication bias generating overestimates of 81 and 55 percent for the median study, respectively.   The range of estimated effects is also somewhat larger than in the case of fixed effects.   The average [minimum, maximum] range is [-6.24,8.25] in the population, compared to [-5.90,8.26] and [-3.50,8.30] in the publication-biased samples.   The average t-statistics for the median study is 0.79 in the population, and 2.55 and 1.28 in the two population-biased samples.   The distribution of precision weights are somewhat more evenly distributed in TABLE 4 compared to TABLE 1.

TABLE 4 shows that our meta-analysis samples are characterized by a substantial amount of non-sampling-error-related heterogeneity, as measured by I.   In the "population" of 300 individual studies, 84% of the variation in the estimated effects is attributed to RE heterogeneity for the median study.   The corresponding percentages for the publication-biased samples are 93% and 79%.   As noted above, these values are typical for economics studies.

FIGURE 2 is the Random Effects analogue to FIGURE 1, showing funnel plots both before and after publication selection. The vertical line now reports the *mean* true effect (=1). The top figure shows the population of effects estimated by the 300 studies. A important difference with previous funnel graphs is immediately apparent: In the Fixed Effects case, the funnel graphs were tightly clustered around the true effect for small SE values. In the Random Effects case, the fact that there are now multiple "true effects" causes the top of the funnel graph to be diffused. The same diffuse pattern is evident in both of the post-publication bias funnel graphs. This has important consequences for the ability of the different MA estimators to correct for publication bias. As in the Fixed Effect case, it is apparent from the funnel graphs that publication selection distorts the distribution of estimates around the mean true effect, inducing a bias.

IIIC. RANDOM EFFECTS: PERFORMANCE TESTS

TABLE 5. TABLE 5 is the Random Effects analogue to TABLE 2. It investigates estimator performance in the presence of publication bias against insignificance. As in the Fixed Effects case, when $\alpha = 0$, the Unadjusted estimator remains an unbiased estimator of the true effect. As α increases, estimation bias at first worsens, then eventually improves when more studies are "published." A major difference with TABLE 2 is that many of the MA estimators produce biased estimates of the mean effect for nonzero values of α. Subsequent Random Effects experiments will produce similar outcomes.

On the dimension of estimation bias, when $0 < \alpha < 2$, the PET performs best of all MA estimators. For $\alpha > 2$, the PEESE and FE/WLS estimators perform best. The RE estimator consistently overestimates the mean true effect for nonzero values of α. For example, when $\alpha = 1$, the RE estimator overestimates the mean true effect by 57 percent.

With respect to MSE performance, the FE/WLS estimator performs as well, or better, than either the PET or the PEESE estimators for all values of α, even though it at times has a

larger bias. It is important to keep in mind that an estimator can be inferior with respect to estimation bias, but superior with respect to MSE performance. We shall have more to say about this below.

Finally, when it comes to hypothesis testing, the lesson from the bottom panel of TABLE 5 could perhaps be summarized as "don't." In almost every case, the Type I error rates are so much larger than 5% that any results from hypothesis testing about the mean true effect should be regarded as highly dubious.

TABLE 6. TABLE 6 repeats the preceding analysis for the case when publication bias discriminates against negative effect estimates. Like previous cases, the Unadjusted estimator produces substantially biased estimates of the mean true effect. All the MA estimators produce biased estimates of the mean effect for small values of α. These biases eventually disappear as the proportion of included samples becomes large.

TABLE 6 tells a story for MSE performance that is similar to TABLE 5. The FE/WLS estimator often performs as well, or better, than either the PET or the PEESE estimators. Interestingly, when $\alpha \geq 2$, the RE estimator is most efficient, despite being the most biased. The explanation has to do with the fact that RE estimates have generally smaller variance than the other MA estimators.[11] This is illustrated in FIGURE 3, which plots the distribution of estimated RE and PEESE estimates when $\alpha = 3$. FIGURE 3 makes the general point that the MA estimators can have substantially different variances, so that a single focus on biasedness can be insufficient when comparing estimator performance.[12,13] Finally, as in TABLE 5, hypothesis testing is highly unreliable and should be avoided.

---

[11] The RE estimator divides the estimated effects by $(SE_i + \tau^2)$, while the other MA estimators divide by SE_i. The effect of adding a large, constant value to SE_i serves to reduce the variation in the weighting term. As a result, the RE estimator will generally have smaller variance than the other MA estimators.

[12] For example, in their response to Mekasha and Tarp (2013), Doucouliagos and Paldam (2013, p. 586) state, "One fundamental difference is that M&T13 strongly advocate the use of the random effects model, whereas D&P08 draw statistical inferences from the fixed effects models. Stanley (2008) and Stanley and Doucouliagos (2012) show that while both fixed and random effects weighted averages are biased in the presence of publication selection, fixed effects averages are less biased (this explains why M&T13 find significantly larger

Summarizing the Random Effects case, we find that the MA estimators that do not explicitly correct for publication bias often perform as well, if not better, than those that do. While the MA estimators always reduce estimation bias in our experiments, they do not always eliminate it – in contrast to the Fixed Effects case. And while the RE estimator is generally the most biased, it sometimes offers substantial efficiency gains over the other MA estimators. Finally, the results with respect to hypothesis testing are generally excessively poor.

Monte Carlo simulations and external validity. A problem with Monte Carlo analyses is that it is difficult to determine if the associated artificial data represent empirical situations that a researcher would actually encounter. We have tried to address that concern in reporting representative sample characteristics for the simulated data in TABLE 1 (Fixed Effects case) and TABLE 4 (Random Effects case). Another approach to determine whether the funnel plots of FIGURES 1 and 2 are realistic is to compare them with funnel plots published in the literature.

FIGURES 4 and 5 present examples of funnel graphs from the MA literature. These can be compared with the funnel graphs of FIGURES 1 and 2. The four funnel plots presented in FIGURE 1 are all taken from Stanley and Doucouliagos's excellent treatise on meta-regression analysis (Stanley and Doucouliagos, 2012). They represent funnel graphs from a variety of different sources, although in each case publication bias is likely not a concern. Accordingly, these would be comparable to Panel A of FIGURE 1.[14]

FIGURE 5 presents a number of funnel graphs that portray the random effects case. Panel A shows little evidence of publication bias and is most comparable to Panel A of FIGURE 2. Panel B shows some evidence of bias against statistical significance and is most

---

meta-averages with the random effects weighted average)." Our results show that just because the FE estimator is less biased than the RE estimator, that does not imply that the associated estimates are "better."

[13] The same phenomenon can be observed for the Unadjusted estimator. For large values of α, the Unadjusted estimator has lower MSE than all but the RE estimator.

[14] Note, however that the scale on the vertical axis in FIGURE 4 is $(1/SE)$ rather than SE, as in FIGURE 1.

comparable to Panel B of FIGURE 2. Panel C shows clear evidence of publication bias against wrong-signed estimates (since this is a price elasticity, positive coefficients would be regarded as "wrong-signed") and is therefore comparable to Panel C of FIGURE 2. Together, these funnel graphs provide additional evidence that the artificial data we have created is similar to data encountered by meta-analysis researchers.

IV. PANEL RANDOM EFFECTS

IVA. PANEL RANDOM EFFECTS: DESIGN OF MONTE CARLO EXPERIMENTS

SIMULATING THE STUDIES. Our last set of experiments examines the performance of the respective MA estimators when studies have multiple estimates, and true effects differ both across and within studies. To the best of our knowledge, this is the only time that Monte Carlo analysis has been used to evaluate studies with multiple estimates in the presence of publication bias.[15]

There is a debate in the literature as to whether MA studies should include all estimates from a study, or just one, or a selected few. To the extent a consensus exists, it is that meta-analysis estimators should include all the estimates, but correct for error correlation across estimates within studies (Stanley and Doucouliagos, 2012; Ringquist, 2013). Accordingly, we do this for the PET and PEESE estimators. In contrast, we do not report clustered standard errors for the other MA estimators. Most researchers are likely to obtain FE, and especially RE, estimates from a canned program such as Stata. Typically, these programs do not allow for clustering in calculating standard errors. As a result, we report estimates for the other MA estimators as researchers are likely to obtain them from canned

---

[15] Bijmolt and Pieters (2001) use Monte Carlo simulations to analyze meta-analysis estimators under single and multiple sampling, but they do not incorporate publication bias.

programs, without adjusted standard errors. We note that our findings are qualitatively unaffected when we do adjust the standard errors for clustering.[16]

The Monte Carlo experiments assume a population of 30 studies, each with 10 estimates per study, where each estimate is based upon 100 observations. We model panel random effects by assuming that estimates within a given study are random, but have the same mean true effect. Estimates from different studies have different mean true effects. Equations (1'.A) and (1'.B) from the Random Effects case are modified accordingly to be:

(1".A) $\alpha_i = \alpha + 2*N(0,1)$, $i = 1,2,\ldots,30$

(1".B) $\alpha_{ij} = \alpha_i + 0.5*N(0,1)$, $j = 1,2,\ldots,10$.

(1".C) $y_{ijt} = 1 + \alpha_{ij} \cdot x_{ij,t} + e_{ijt}$, $t = 1,2,\ldots,100$.

The different weights on the standard normal variates in (1".A) and (1".B) are designed to capture the idea that effects are more likely to be similar within a study than across studies.

We model the error terms similarly, with error variances differing both within and across studies, but with most of the variation occurring across studies. Equations (3.A) and (3'.B) are modified from the Random Effects case to be:

(3".A) $e_{ijt} = \lambda_{ij} \cdot NID(0,1)$, where

(3".B1) $\lambda_i = 0.5 + 30*UID(0,1)$, and

(3".B2) $\lambda_{ij} = \lambda_i + UID(0,1)$.

As in the RE case, these DGP parameters are designed to simultaneously the four criteria listed above.

A second difference with previous studies concerns publication bias. Our experiments assume that the bias works at the level of the study, and not the individual estimate. In the case of bias against statistical insignificance, we assume that in order to be published, a study must have most of its estimates (7 out of 10, or more) be statistically

---

[16] For example, the Type I error rates in TABLE 8 for the WLS estimator fall from approximately 80 percent to approximately 25 percent when standard errors are clustered. While this is a large decrease, it does not qualitatively change the conclusions we draw from these experiments.

significant. If the study meets that selection criterion, all the estimates from that study are "published." If the study does not meet that criterion, none of the estimates from that study are published. An identical "7 out of 10, or more" rule applies to publication bias against wrong-signed estimates.

A third difference has to do with the specification of our MA regressions. We modify Equation (4) to include multiple estimates per study, and allow study-specific fixed effects in the error term:

(10) $\hat{\beta}_{ij1} = \alpha + \lambda_i + v_{ij},$

where $\lambda_i \equiv \alpha - \alpha_i$ (cf. Equation 1''.A). Dividing through by the appropriate standard error (either SE_{ij} or $\omega_{ij} = \sqrt{(SE_{ij})^2 + \tau^2}$) produces the FE, WLS, and RE estimators as described above.

For the PET estimator, we model our specification after the recommendations of Stanley and Doucouliagos (2012, see (i) Equation 5.5, p. 85, and (ii) Equation 5.9, p. 101) to obtain the following "panel fixed effects" specification for the PET estimator when studies have more than one estimate:

(11) $\hat{\beta}_{ij1} = \alpha + \sum_i \gamma_i SE_{ij} D_{ij} + \lambda_i + v_{ij},$

where D_{ij} is a dummy variable that takes the value 1 for study i and 0 for other studies, and noting again that $\lambda_i \equiv \alpha - \alpha_i$. Dividing through by SE_{ij} produces the following specification:

(12) $\dfrac{\hat{\beta}_{ij1}}{SE_{ij}} = \alpha \left(\dfrac{1}{SE_{ij}}\right) + \sum_i \gamma_i D_{ij} + \dfrac{(\lambda_i + v_{ij})}{SE_{ij}}$.

The panel version of the PEESE estimator is given by:

(13) $\dfrac{\hat{\beta}_{ij1}}{SE_{ij}} = \alpha + \sum_i \gamma_i SE_{ij} D_{ij} + \dfrac{(\lambda_i + v_{ij})}{SE_{ij}}$.

Coefficient standard errors are adjusted to allow for within-study correlation of error terms.

IVB. PANEL RE: AN EXAMPLE META-ANALYSIS SAMPLE ($\alpha = 1$)

TABLE 7 does the same thing for the Panel Random Effects case that TABLES 1 and 4 did for the Fixed Effects and Random Effects cases. It reports average characteristics of an example MA sample with an *overall mean* true effect of one ($\alpha = 1$). A total of 30 studies (= 300 effect estimates since each study has 10 estimates) is included in the population of studies prior to publication bias. An average of only 21.8 and 56.2 percent of studies are included in the meta-analysis study due to the two types of publication bias. For this particular example, the respective publication biases generate estimated effect biases of 124% for the median study when publishing is biased against statistical insignificance, and 122% when the bias is against "wrong-signed" (= negative) estimates.

The range of estimated effects and *t*-statistics is somewhat narrower than in previous cases, while the median *t*-statistics for the two publication-biased samples are somewhat larger. The measure of effect heterogeneity, I^2, is approximately the same as in the Random Effects case.

FIGURE 6 shows corresponding funnel graphs for the pre- and post-publication bias samples for this example. The vertical line reports the overall mean of true effects across estimates and studies ($\alpha = 1$). The top figure shows the population of 300 estimated effects. We know from the preceding analysis that the degree of diffusion at the top of the funnel will be problematic for the MA estimators. The task of identifying the mean true effect is hampered when there exists a wide range of estimated effects with small standard errors.

The other noteworthy characteristic of FIGURE 6, which is particularly evident in Panel B, is the degree of clustering in the estimates. Because all of the estimates from a given study are published, even if some individual estimates do not satisfy the publication criterion, a given study's estimates cluster together. This emphasizes the importance of controlling for lack of independence across estimates from the same study. As in previous

cases, the distorting effects of publication bias are clearly evident in the two post-publication bias funnel graphs.

IVC. PANEL RANDOM EFFECTS: PERFORMANCE TESTS

TABLE 8. As we have seen in previous cases, the Unadjusted estimator provides an unbiased estimate of the mean true effect when publication bias discriminates against statistical insignificance and $\alpha = 0$. As α increases, publication bias at first worsens, then eventually improves when more studies are "published." The estimation bias for the Unadjusted estimator in the Panel Random Effects case can be quite substantial. Consistent with TABLE 5, the RE estimator is positively biased for positive α. The major difference between TABLE 8 and TABLE 5 is that the other MA estimators now also show substantial bias. For example, when $\alpha = 2.0$, the average estimated effects for the PET, PEESE, and FE/WLS estimators are 2.41, 2.42, and 2.50. These are, however, much better than the Unadjusted and RE values of 3.37 and 3.10, respectively.

Superiority in unbiasedness does not necessarily translate into superiority in MSE performance. Despite having larger bias, the RE estimator always has lower MSE than the PET and PEESE estimators, and does better than the FE/WLS estimator for most values of α. This seeming anomaly was previously addressed in the discussion around FIGURE 3. And, as we also saw in the Random Effects case, the performance of all the estimators are extremely poor when it comes to hypothesis testing. Type I error rates for all of the MA estimators substantially exceed their expected values of 5 percent.

TABLE 9. When publication bias is directed against wrong-signed estimates, the story is much the same. TABLE 9 provides further evidence of the positive bias of all five of the meta-analysis estimators. While the MA estimators always produce estimates that are less biased than the Unadjusted estimator, the reduction in estimation bias is often quite small. For example, $\alpha = 2.0$, the Unadjusted estimator produces a mean estimate of 2.90,

while the PET, PEESE, and FE/WLS estimators produce estimates of 2.63, 2.64, and 2.66, respectively. The bias in estimated effects disappears only as publication bias becomes less severe and more studies are included in the meta-analysis sample. In contrast to TABLE 8, the RE estimator is now either best or close to best on the dimension of MSE for all values of α. It is worth noting that the Unadjusted estimator often has lower MSE than all but the RE estimator, and even dominates the RE estimator when α = 4. Reliability in hypothesis testing for all the estimators continues to be abysmal across the full range of α values.

The overall poor performance of the respective MA estimators is graphically represented in FIGURE 7. The figure represents the case when α = 1 and publication bias discriminates against wrong-signed estimates. As in previous figures, the solid vertical line represents the mean true effect in the population. The Unadjusted estimator demonstrates the greatest bias, followed closely by the RE estimator, which only marginally improves on the OLS estimator in this case. The other four MA estimators (PET, PEESE, and FE/WLS), are less biased, but are also more dispersed.

V. CONCLUSION

This study uses Monte Carlo analysis to investigate the performances of five different meta-analysis (MA) estimators: the Precision Effect Test (PET) estimator, the Precision Effect Estimate with Standard Errors (PEESE) estimator, the Fixed Effects (FE) estimator, the Weighted Least Squares (WLS) estimator, and the Random Effects (RE) estimator. We consider two types of publication bias: publication bias directed against statistically insignificant estimates, and publication bias directed against wrong-signed estimates. Finally, we consider three cases concerning the distribution of the "true effect": the Fixed Effects case, where there is only estimate per study, and all studies have the same true effect; the Random Effects case, where there is only one estimate per study, and there is

heterogeneity in true effects across studies; and the Panel Random Effects case, where studies have multiple estimates, and true effects are heterogeneous both across and within studies.

While our simulations employ a specific set of DGP parameter values, they allow the following four conclusions. First, MA estimators that correct for publication bias (the PET and PEESE estimators) sometimes perform worse than those that do not correct for publication bias. Second, when publication bias exists and there is substantial heterogeneity in true effects across studies, MA estimators may only moderately reduce publication bias. Third, while the RE estimator is often more biased than other MA estimators, it sometimes is the most efficient. And finally, hypothesis testing about the mean true effect is generally unreliable for all MA estimators.

These findings lead us to make two recommendations to meta-analysts interested in identifying "true effects" from publication-biased samples. The first is that meta-analysts should use a variety of MA estimators in their research, not just one or two. The second is that hypothesis testing about true mean effects should be avoided because it is too unreliable.

More specifically, our results suggest caution in employing the FAT-PET-PEESE procedure for estimating effects in the presence of publication bias (cf. Stanley and Doucouliagos, 2012, pages78f.). This approach advocates that researchers follow the following four-step procedure:

- STEP ONE: Test for publication bias using the PET specification;

- STEP TWO: Test whether there is a nonzero mean effect using the PET specification;

- STEP THREE: If one fails to reject the null hypothesis of no effect in STEP TWO, conclude that there is no evidence of an empirical effect.

- STEP FOUR: If one rejects the null hypothesis of no effect in STEP TWO, estimate the size of the effect using the PEESE specification.

The findings of unreliability of hypothesis testing and the lack of general superiority of any one MA estimation in our Monte Carlo simulations suggest that there is insufficient evidence at this time to limit meta-analytic statistical inference to this approach.

A final takeaway from our study is that much work remains to be done on meta-analysis. While current meta-analysis procedures do, for the most part, offer improvements on naïve averaging of published estimates, it is clear from the findings reported here that there is much room for improvement. It is hoped that this research will stimulate future efforts in this direction.

REFERENCES

Bellavance, R., Dionne, G., and Lebeau, M. "The value of a statistical life: A meta-analysis with a mixed effects regression model." *Journal of Health Economics*, Vol. 28 (2009): 444-64.

Bijmolt, T.H.A. and R.G.M. Pieters. "Meta-analysis in marketing when studies contain multiple measurements." *Marketing Letters*, Vol. 12 (2001), 157-69.

Dalhuisen, J.M., Florax, R.J.G.M., de Groot, H.L.F., and Nijkamp, P. "Price and income elasticities of residential water demand: A meta-analysis." *Land Economics*, Vol. 79 (2003): 292-308.

Doucouliagos, H. and Paldam, M. "The robust result in meta-analysis of aid effectiveness: A response to Mekasha and Tarp." *The Journal of Development Studies*, Vol. 49, No. 4 (2013): 584-587.

Doucouliagos, H. and Stanley, T.D. "Publication selection bias in minimum wage research? A meta-regression analysis." *British Journal of Industrial Relations*, Vol. 47 (2009): 406-28.

Higgins, J.P.T., and Thompson, S.G. "Quantifying heterogeneity in a meta-analysis." *Statistics in Medicine*, Vol. 21 (2002): 1539-1558.

Higgins, J.P.T., Thompson, S.G., Deeks, J.J., and Altman, D.G. "Measuring inconsistency in meta-analyses." *British Medical Journal*, Vol. 327 (2003): 557-560.

Koetse, M.J., Florax, R.J.G.M., and de Groot, H.L.F. "Consequences of effect size heterogeneity for meta-analysis: a Monte Carlo study." *Statistical Methods and Applications*, Vol. 19 (2010): 217-236.

Mekasha, T.J. and Tarp, F. "Aid and growth: What meta-analysis reveals." *The Journal of Development Studies,* Vol. 49, No. 4 (2013): 564-583.

Moreno, S.G., Sutton, A.J., Ades, A.E. Stanley, T.D., Abrams, K.R., Peters, J.L., and Cooper, N.J. "Assessment of regression-based methods to adjust for publication bias through a comprehensive simulation study." *BMC Medical Research Methodology* (2009) 9:2, http://www.biomedcentral.com/1471-2288/9/2

Nelson, J.P. "Estimating the price elasticity of beer: Meta-analysis of data with heterogeneity, dependence, and publication bias." *Journal of Health Economics*, Vol. 33 (2014): 180-187.

Ringquist, E.J. *Meta-analysis for public management and policy.* San Franciso: Jossey-Bass, 2013.

Stanley, T.D. "Meta-regression methods for detecting and estimating empirical effects in the presence of publication selection." *Oxford Bulletin of Economics and Statistics*, Vol. 70, Issue 1 (2008): 103-127.

Stanley, T.D., and Doucouliagos, H. *Meta-regression analysis in economics and business.* London: Routledge, 2012.

Stanley, T.D., and Doucouliagos, H. "Better than random: Weighted least squares meta-regression analysis". Working paper, Deakin University, School of Accounting, Economics, and Finance, Economics Series, SWP 2013/2, updated February 2014(a).

Stanley, T.D., and Doucouliagos, H. "Neither fixed nor random: Weighted least squares meta-analysis". Working paper, Deakin University, School of Accounting, Economics, and Finance, Economics Series, SWP 2013/1, updated March 2014(b).

Stanley, T.D., Jarrell, S.B. and Doucouliagos, C. "Could it be better to discard 90% of the data? A statistical paradox." *American Statistician* Vol. 64 (2010): 70-77.

TABLE 1
Sample Characteristics for a Simulated Meta-Analysis Data Set: Fixed Effects ($\alpha = 1$)

Variable	Median	Minimum	P5%	P95%	Maximum
PRE-PUBLICATION BIAS (100 percent of estimates):					
Estimated effect	1.00	-5.74	-1.98	4.00	7.76
t-statistic	0.94	-2.31	-0.95	6.23	36.69
Precision (1/SE)	0.66	0.28	0.34	6.04	36.63
PUBLICATION BIAS AGAINST INSIGNIFICANCE (31.8 percent of estimates):					
Estimated effect	1.19	-5.01	-0.93	5.13	7.65
t-statistic	2.62	-2.15	-0.47	15.19	36.23
Precision (1/SE)	1.66	0.31	0.37	15.11	36.19
PUBLICATION BIAS AGAINST NEGATIVE EFFECTS (80.6 percent of estimates):					
Estimated effect	1.21	-3.10	0.12	4.25	7.72
t-statistic	1.27	-1.37	0.08	7.38	36.93
Precision (1/SE)	0.76	0.28	0.34	7.24	36.89

FIGURE 1
Example of Funnel Graphs for a Simulated Meta-Analysis Data Set: Fixed Effects

A. Pre-Publication Bias
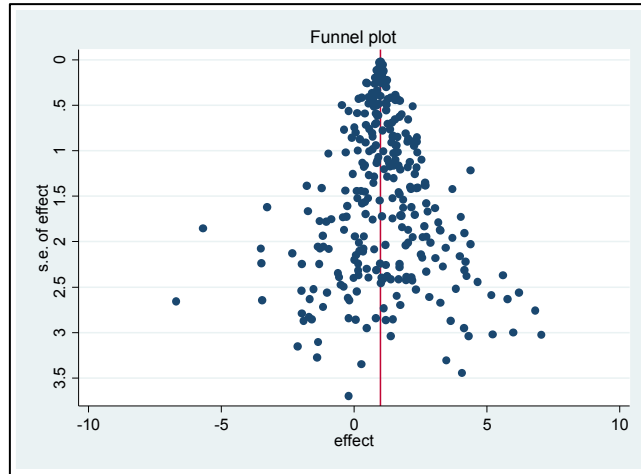
B. Post-Publication Bias (against statistical insignificance)
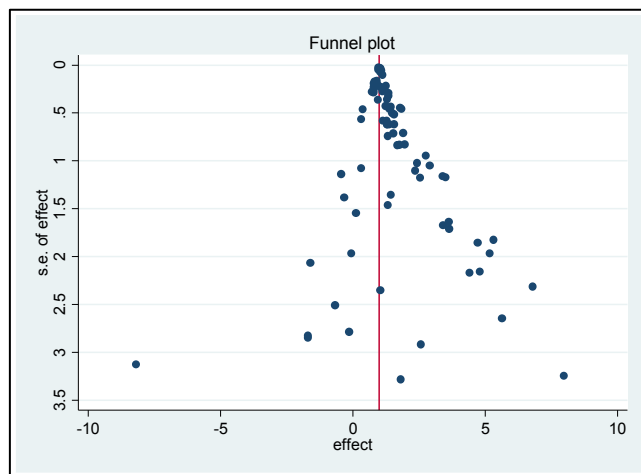
C. Post-Publication Bias (against wrong-signed estimates)

TABLE 2
Comparative Performance of Meta-Analysis Estimators:
Fixed Effects/Publication Bias against Insignificance

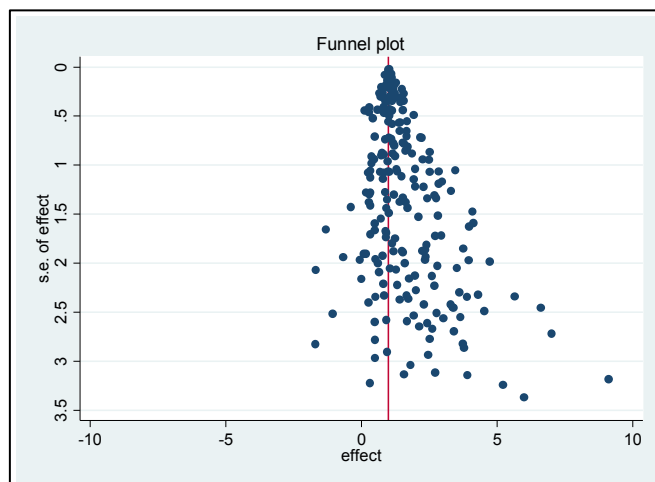| α	Percent	Unadjusted	PET	PEESE	FE	WLS	RE
			Mean Estimate of True Effect				
0.0	14.4	0.02	0.00	0.00	0.00	0.00	0.00
0.5	23.0	0.91	0.48	0.51	0.51	0.51	0.56
1.0	31.7	1.57	0.96	1.01	1.01	1.01	1.05
1.5	40.0	2.14	1.46	1.50	1.51	1.51	1.54
2.0	47.5	2.67	1.96	2.00	2.01	2.01	2.04
2.5	54.6	3.17	2.46	2.49	2.51	2.51	2.53
3.0	61.1	3.65	2.97	2.99	3.01	3.01	3.02
3.5	66.9	4.12	3.47	3.49	3.51	3.51	3.52
4.0	72.4	4.57	3.97	3.99	4.01	4.01	4.01
			Mean Squared Error				
0.0	14.4	0.1827	0.0227	0.0134	0.0108	0.0108	0.0365
0.5	23.0	0.2419	0.0010	0.0005	0.0006	0.0006	0.0053
1.0	31.7	0.3683	0.0018	0.0003	0.0005	0.0005	0.0040
1.5	40.0	0.4317	0.0021	0.0003	0.0004	0.0004	0.0027
2.0	47.5	0.4667	0.0021	0.0003	0.0004	0.0004	0.0021
2.5	54.6	0.4623	0.0020	0.0003	0.0004	0.0004	0.0016
3.0	61.1	0.4347	0.0018	0.0003	0.0003	0.0003	0.0010
3.5	66.9	0.3944	0.0015	0.0003	0.0003	0.0003	0.0007
4.0	72.4	0.3404	0.0013	0.0003	0.0003	0.0003	0.0005
			Type I Error Rates				
0.0	0.06	0.25	0.25	0.23	0.04	0.02	0.06
0.5	0.39	0.25	0.18	0.16	0.06	0.13	0.39
1.0	0.84	0.45	0.11	0.14	0.07	0.18	0.84
1.5	0.99	0.53	0.10	0.12	0.07	0.15	0.99
2.0	1.00	0.50	0.12	0.13	0.07	0.14	1.00
2.5	1.00	0.52	0.11	0.11	0.08	0.12	1.00
3.0	1.00	0.48	0.13	0.10	0.08	0.08	1.00
3.5	1.00	0.44	0.14	0.09	0.09	0.08	1.00
4.0	1.00	0.40	0.15	0.07	0.07	0.06	1.00

TABLE 3
Comparative Performance of Meta-Analysis Estimators:
Fixed Effects/Publication Bias against Wrong Sign

α	Percent	Unadjusted	PET	PEESE	FE	WLS	RE
			Mean Estimate of True Effect				
0.0	55.1	1.01	0.00	0.06	0.09	0.09	0.11
0.5	71.7	1.21	0.47	0.50	0.52	0.52	0.52
1.0	80.6	1.56	0.97	1.00	1.01	1.01	1.01
1.5	86.5	1.94	1.48	1.50	1.51	1.51	1.51
2.0	90.6	2.35	1.98	2.00	2.00	2.00	2.01
2.5	93.5	2.76	2.49	2.50	2.50	2.50	2.50
3.0	95.5	3.20	2.99	3.00	3.00	3.00	3.00
3.5	96.9	3.65	3.49	3.50	3.50	3.50	3.50
4.0	98.0	4.10	4.00	4.00	4.00	4.00	4.00
			Mean Squared Error				
0.0	55.1	1.0280	0.0004	0.0049	0.0099	0.0099	0.0156
0.5	71.7	0.5201	0.0015	0.0003	0.0006	0.0006	0.0007
1.0	80.6	0.3242	0.0011	0.0003	0.0004	0.0004	0.0004
1.5	86.5	0.1985	0.0007	0.0002	0.0003	0.0003	0.0003
2.0	90.6	0.1274	0.0007	0.0003	0.0003	0.0003	0.0003
2.5	93.5	0.0754	0.0005	0.0003	0.0003	0.0003	0.0003
3.0	95.5	0.0465	0.0004	0.0003	0.0003	0.0003	0.0003
3.5	96.9	0.0319	0.0004	0.0003	0.0003	0.0003	0.0003
4.0	98.0	0.0200	0.0003	0.0002	0.0002	0.0002	0.0003
			Type I Error Rates				
0.0	55.1	1.00	0.14	0.81	0.97	0.97	0.98
0.5	71.7	1.00	0.49	0.11	0.20	0.25	0.20
1.0	80.6	1.00	0.42	0.11	0.11	0.14	0.10
1.5	86.5	1.00	0.27	0.09	0.08	0.11	0.07
2.0	90.6	0.99	0.25	0.12	0.07	0.09	0.07
2.5	93.5	0.86	0.18	0.11	0.07	0.08	0.06
3.0	95.5	0.56	0.14	0.10	0.05	0.06	0.05
3.5	96.9	0.36	0.14	0.12	0.07	0.08	0.06
4.0	98.0	0.19	0.11	0.09	0.04	0.05	0.04

TABLE 4
Sample Characteristics for a Simulated Meta-Analysis Data Set: Random Effects ($\alpha = 1$)

Variable	Median	Minimum	P5%	P95%	Maximum
PRE-PUBLICATION BIAS (100 percent of estimates):					
Estimated effect	1.00	-6.24	-2.38	4.37	8.25
t-statistic	0.79	-8.12	-1.48	5.98	31.79
Precision (1/SE)	0.65	0.28	0.33	5.06	17.66
I2	0.84	0.65	0.74	0.92	0.95
PUBLICATION BIAS AGAINST INSIGNIFICANCE (32.9 percent of estimates):					
Estimated effect	1.81	-5.90	-2.09	5.64	8.26
t-statistic	2.55	-8.15	-2.29	12.77	31.50
Precision (1/SE)	1.25	0.31	0.37	9.63	17.26
I2	0.93	0.72	0.87	0.97	0.99
PUBLICATION BIAS AGAINST NEGATIVE EFFECTS (74.7 percent of estimates):					
Estimated effect	1.55	-3.50	0.01	4.76	8.30
t-statistic	1.28	-2.93	0.01	7.40	31.57
Precision (1/SE)	0.70	0.28	0.34	5.57	17.03
I2	0.79	0.41	0.63	0.90	0.95

FIGURE 2
Example of Funnel Graphs for a Simulated Meta-Analysis Data Set: Random Effects

A. Pre-Publication Bias
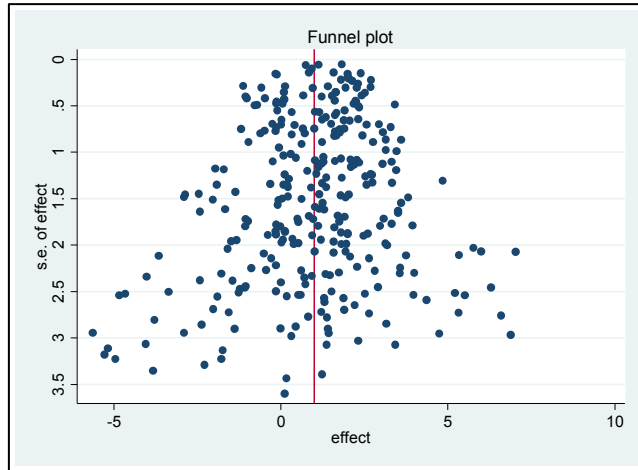
B. Post-Publication Bias (against statistical insignificance)
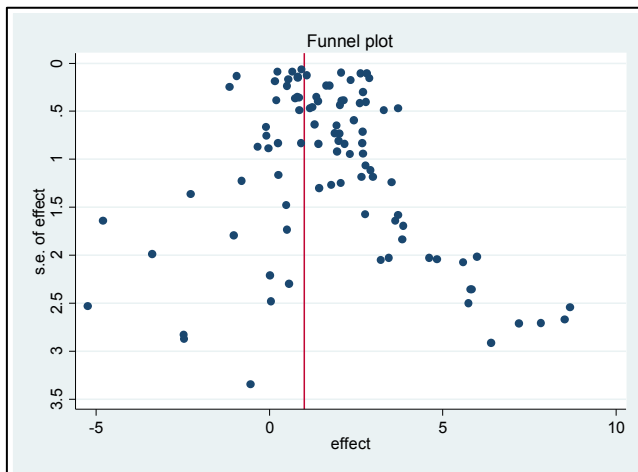
C. Post-Publication Bias (against wrong-signed estimates)

TABLE 5
Comparative Performance of Meta-Analysis Estimators:
Random Effects/Publication Bias against Insignificance

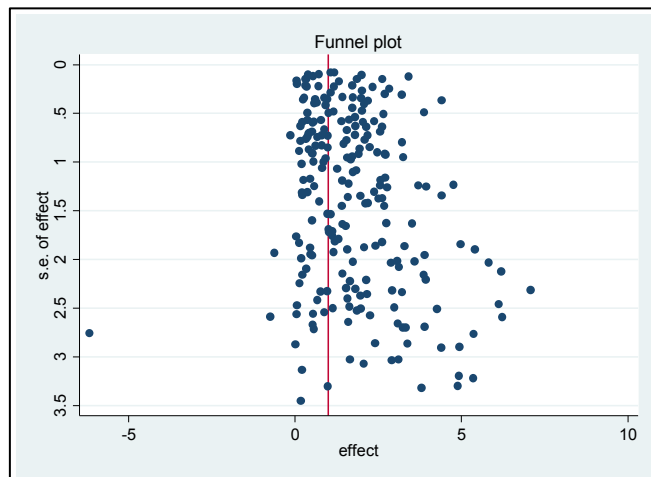| α	Percent	Unadjusted	PET	PEESE	FE	WLS	RE
			Mean Estimate of True Effect				
0.0	27.1	0.01	0.00	0.00	0.00	0.00	0.01
0.5	28.8	1.00	0.51	0.59	0.60	0.60	0.88
1.0	33.2	1.82	1.04	1.14	1.17	1.17	1.57
1.5	39.2	2.44	1.51	1.62	1.65	1.65	2.08
2.0	46.0	2.94	1.93	2.05	2.09	2.09	2.53
2.5	52.7	3.40	2.41	2.51	2.55	2.55	2.95
3.0	59.1	3.84	2.91	3.00	3.04	3.04	3.39
3.5	65.1	4.28	3.41	3.49	3.53	3.53	3.84
4.0	70.4	4.71	3.95	4.00	4.04	4.04	4.29
			Mean Squared Error				
0.0	27.1	0.0871	0.1749	0.1089	0.1032	0.1032	0.0437
0.5	28.8	0.3272	0.1840	0.1230	0.1192	0.1192	0.1822
1.0	33.2	0.7382	0.1649	0.1220	0.1232	0.1232	0.3522
1.5	39.2	0.9210	0.1366	0.0990	0.1027	0.1027	0.3605
2.0	46.0	0.9117	0.1343	0.0811	0.0806	0.0806	0.2916
2.5	52.7	0.8252	0.1375	0.0787	0.0750	0.0750	0.2187
3.0	59.1	0.7246	0.1357	0.0807	0.0753	0.0753	0.1649
3.5	65.1	0.6272	0.1420	0.0851	0.0780	0.0780	0.1298
4.0	70.4	0.5222	0.1164	0.0733	0.0682	0.0682	0.0944
			Type I Error Rates				
0.0	27.1	0.04	0.12	0.10	0.88	0.44	0.04
0.5	28.8	0.43	0.16	0.15	0.88	0.51	0.49
1.0	33.2	0.92	0.15	0.17	0.88	0.58	0.93
1.5	39.2	1.00	0.14	0.14	0.89	0.59	0.99
2.0	46.0	1.00	0.15	0.12	0.87	0.57	1.00
2.5	52.7	1.00	0.14	0.11	0.86	0.58	0.98
3.0	59.1	1.00	0.15	0.12	0.86	0.60	0.96
3.5	65.1	1.00	0.14	0.12	0.89	0.62	0.92
4.0	70.4	1.00	0.12	0.10	0.86	0.60	0.83

TABLE 6
Comparative Performance of Meta-Analysis Estimators:
Random Effects/Publication Bias against Wrong Sign

α	Percent	Unadjusted	PET	PEESE	FE	WLS	RE
			Mean Estimate of True Effect				
0.0	55.1	1.26	0.60	0.66	0.69	0.69	0.90
0.5	65.5	1.53	0.89	0.94	0.97	0.97	1.18
1.0	74.5	1.81	1.21	1.26	1.29	1.29	1.49
1.5	82.0	2.12	1.58	1.62	1.64	1.64	1.84
2.0	87.5	2.48	2.01	2.04	2.06	2.06	2.23
2.5	91.3	2.86	2.49	2.51	2.53	2.53	2.66
3.0	93.9	3.27	2.96	2.99	3.00	3.00	3.11
3.5	95.9	3.70	3.49	3.50	3.51	3.51	3.58
4.0	97.2	4.15	3.98	3.99	4.00	4.00	4.05
			Mean Squared Error				
0.0	55.1	1.6156	0.4676	0.5072	0.5436	0.5436	0.8234
0.5	65.5	1.0664	0.2415	0.2583	0.2795	0.2795	0.4696
1.0	74.5	0.6631	0.1304	0.1272	0.1367	0.1367	0.2499
1.5	82.0	0.3915	0.0959	0.0749	0.0758	0.0758	0.1215
2.0	87.5	0.2375	0.1081	0.0749	0.0709	0.0709	0.0635
2.5	91.3	0.1400	0.1064	0.0721	0.0665	0.0665	0.0347
3.0	93.9	0.0852	0.1099	0.0736	0.0668	0.0668	0.0211
3.5	95.9	0.0518	0.1127	0.0763	0.0692	0.0692	0.0138
4.0	97.2	0.0370	0.1075	0.0735	0.0668	0.0668	0.0116
			Type I Error Rates				
0.0	55.1	1.00	0.75	0.85	1.00	0.98	1.00
0.5	65.5	1.00	0.42	0.60	0.98	0.95	1.00
1.0	74.5	1.00	0.21	0.31	0.93	0.82	1.00
1.5	82.0	1.00	0.14	0.15	0.84	0.66	0.96
2.0	87.5	1.00	0.13	0.12	0.86	0.65	0.74
2.5	91.3	0.93	0.12	0.11	0.86	0.67	0.44
3.0	93.9	0.73	0.12	0.10	0.86	0.65	0.24
3.5	95.9	0.45	0.13	0.12	0.85	0.65	0.13
4.0	97.2	0.28	0.13	0.11	0.86	0.63	0.09

FIGURE 3
Distribution of RE and PEESE Estimates in the
RE Case, α=3, Publication Bias against Wrong Sign

FIGURE 4
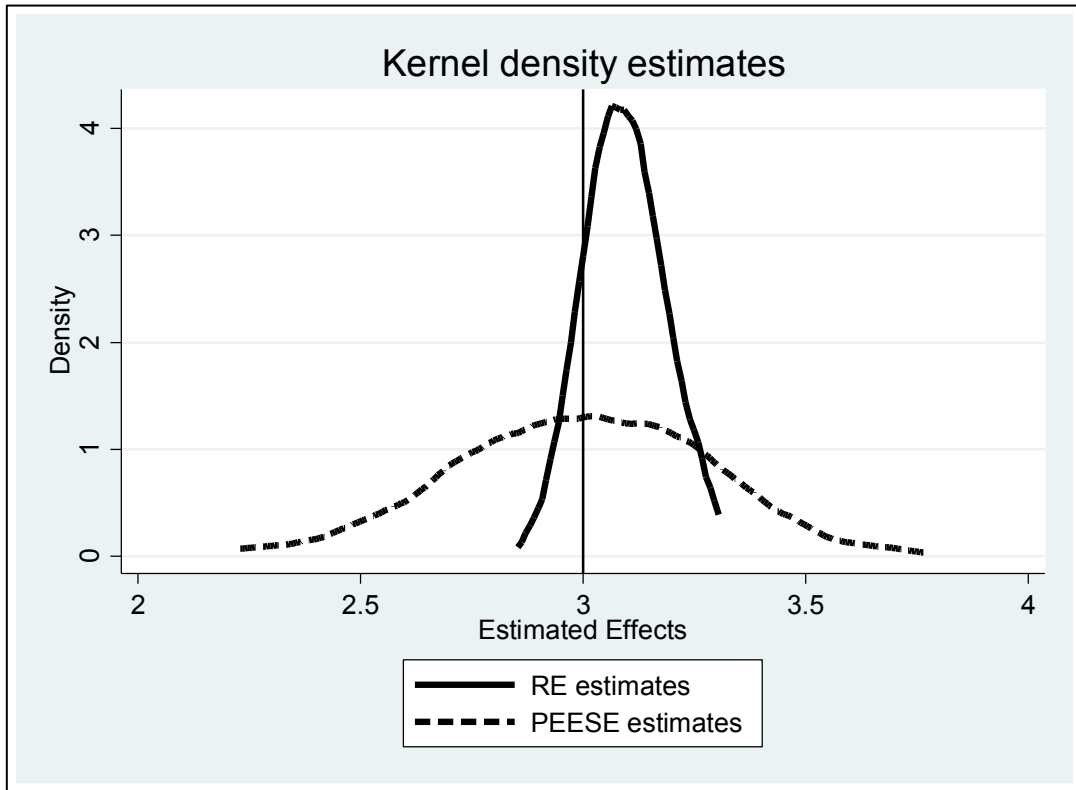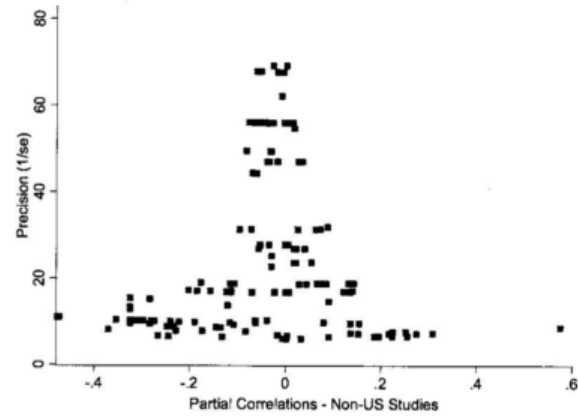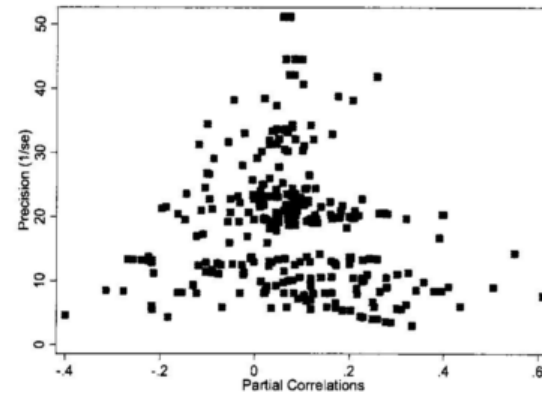**Examples of Funnel Graphs from the Meta-Analysis Literature (Fixed Effects)**

(a) Unions and profits, non-US studies

(b) Aid allocations and democracy

(c) FDI and growth
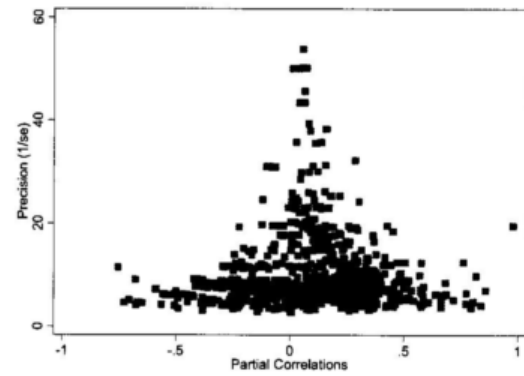
(d) Hospital ownership and costs
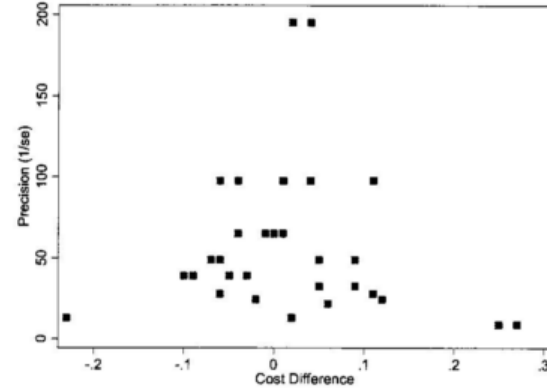
SOURCE: Stanley & Doucouliagos (2012, Figure 4.5, page 58)

A. Mekasha and Tarp (2013, Figure 1, page 571)
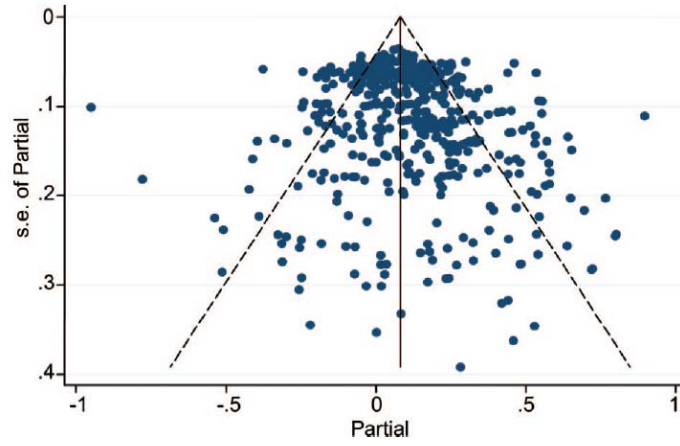
B. Ringquist (2013, Figure 6.7, page 252)
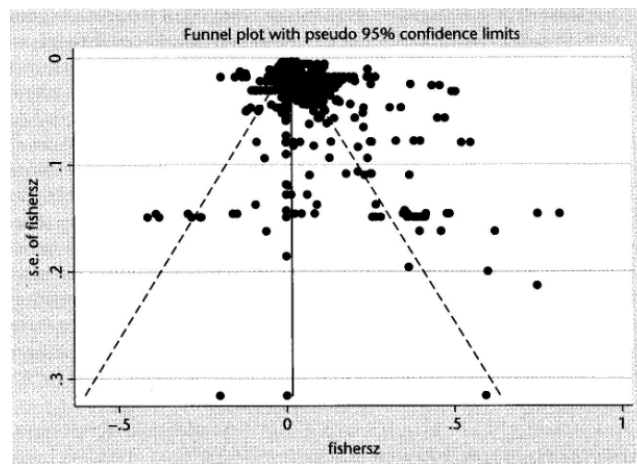
C. Nelson (2014, Figure 1, page 184)
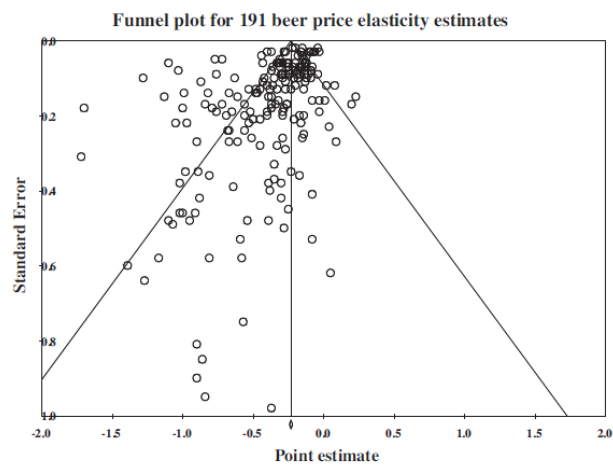
TABLE 7
Sample Characteristics for a Simulated Meta-Analysis Data Set: Panel Data/Random Effects ($\alpha = 1$)

Variable	Median	Minimum	P5%	P95%	Maximum
PRE-PUBLICATION BIAS (100 percent of estimates):					
Estimated effect	0.96	-7.50	-3.47	5.47	9.43
t-statistic	0.67	-10.16	-3.17	7.44	20.69
Precision (1/SE)	0.65	0.27	0.33	4.05	8.82
I2	0.86	0.46	0.68	0.97	0.99
PUBLICATION BIAS AGAINST INSIGNIFICANCE (21.8 percent of estimates):					
Estimated effect	2.24	-3.03	-2.18	5.73	7.03
t-statistic	3.62	-8.81	-6.08	15.56	20.66
Precision (1/SE)	1.98	0.70	0.77	6.62	8.37
I2	0.92	0.00	0.71	0.99	1.00
PUBLICATION BIAS AGAINST NEGATIVE EFFECTS (56.2 percent of estimates):					
Estimated effect	2.22	-3.99	-0.84	6.17	9.33
t-statistic	1.75	-2.11	-0.50	11.28	20.66
Precision (1/SE)	0.72	0.29	0.34	4.49	7.48
I2	0.74	0.11	0.43	0.94	0.98

FIGURE 6
Example of Funnel Graphs for a Simulated Meta-Analysis Data Set:
Panel Random Effects

A. Pre-Publication Bias
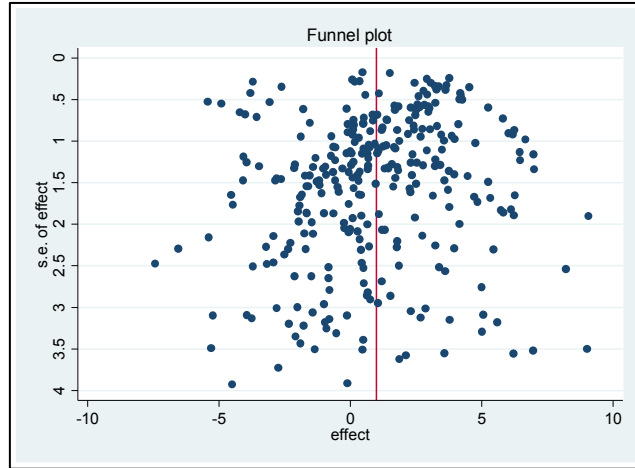
B. Post-Publication Bias (against statistical insignificance)
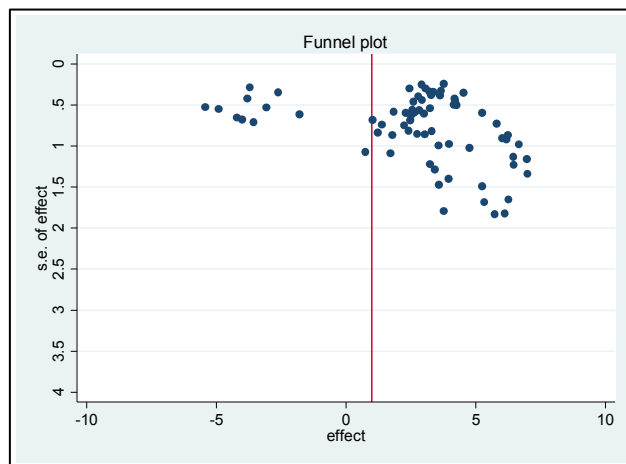
C. Post-Publication Bias (against wrong-signed estimates)
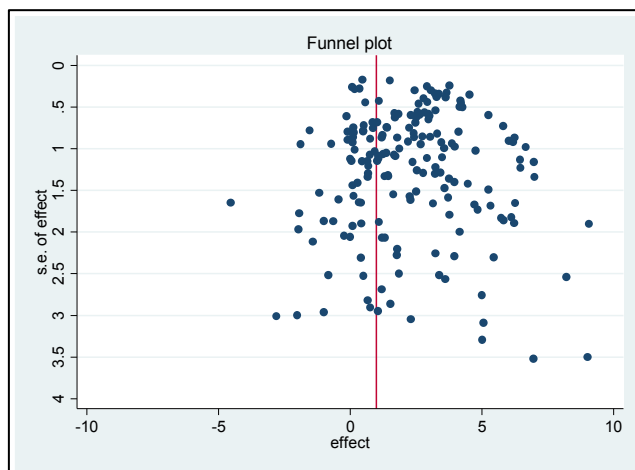
TABLE 8
Comparative Performance of Meta-Analysis Estimators:
Panel Random Effects/Publication Bias against Insignificance

α	Percent	Unadjusted	PET	PEESE	FE	WLS	RE
			Mean Estimate of True Effect				
0.0	19.4	0.01	-0.05	-0.03	-0.03	-0.03	0.00
0.5	19.5	1.11	0.64	0.65	0.70	0.70	1.01
1.0	22.0	2.05	1.36	1.37	1.41	1.41	1.88
1.5	25.3	2.81	1.86	1.86	1.94	1.94	2.59
2.0	29.7	3.37	2.41	2.42	2.50	2.50	3.10
2.5	34.1	3.86	2.90	2.90	2.96	2.96	3.57
3.0	40.4	4.23	3.27	3.27	3.34	3.34	3.94
3.5	46.4	4.66	3.71	3.71	3.79	3.79	4.36
4.0	52.7	5.03	4.16	4.16	4.23	4.23	4.74
			Mean Squared Error				
0.0	19.4	1.951	3.565	2.967	2.289	2.289	1.683
0.5	19.5	2.058	3.646	3.071	2.362	2.362	1.773
1.0	22.0	2.328	3.580	2.918	2.188	2.188	1.870
1.5	25.3	2.505	3.505	2.897	2.060	2.060	1.907
2.0	29.7	2.397	2.773	2.266	1.657	1.657	1.703
2.5	34.1	2.231	2.430	2.018	1.459	1.459	1.547
3.0	40.4	1.828	2.410	1.980	1.314	1.314	1.211
3.5	46.4	1.597	2.341	1.945	1.332	1.332	1.024
4.0	52.7	1.277	2.193	1.805	1.145	1.145	0.799
			Type I Error Rates				
0.0	19.4	0.55	0.38	0.40	0.96	0.76	0.56
0.5	19.5	0.62	0.36	0.38	0.95	0.77	0.61
1.0	22.0	0.69	0.40	0.41	0.96	0.79	0.66
1.5	25.3	0.81	0.41	0.42	0.95	0.79	0.78
2.0	29.7	0.88	0.41	0.43	0.96	0.81	0.83
2.5	34.1	0.92	0.42	0.43	0.94	0.81	0.86
3.0	40.4	0.93	0.41	0.41	0.95	0.81	0.87
3.5	46.4	0.95	0.39	0.40	0.96	0.82	0.85
4.0	52.7	0.94	0.39	0.39	0.93	0.79	0.83

TABLE 9
Comparative Performance of Meta-Analysis Estimators:
Panel Random Effects /Publication Bias against Wrong Sign

α	*N*	*Unadjusted*	*PET*	*PEESE*	*FE*	*WLS*	*RE*
			Mean Estimate of True Effect				
0.0	37.5	2.02	1.72	1.73	1.77	1.77	1.86
0.5	47.9	2.19	1.89	1.90	1.94	1.94	2.04
1.0	56.8	2.41	2.16	2.16	2.17	2.17	2.29
1.5	65.6	2.64	2.41	2.42	2.43	2.43	2.53
2.0	73.7	2.90	2.63	2.64	2.66	2.66	2.79
2.5	80.7	3.17	3.00	3.00	3.00	3.00	3.09
3.0	86.0	3.51	3.30	3.30	3.31	3.31	3.43
3.5	90.4	3.88	3.74	3.73	3.74	3.74	3.82
4.0	93.5	4.27	4.13	4.13	4.14	4.14	4.22
			Mean Squared Error				
0.0	37.5	4.2298	4.7255	3.9699	3.6705	3.6705	3.6813
0.5	47.9	2.9717	3.5415	2.9473	2.6395	2.6395	2.5511
1.0	56.8	2.1057	2.8652	2.3275	1.9743	1.9743	1.8267
1.5	65.6	1.4099	2.5001	2.0126	1.5324	1.5324	1.2030
2.0	73.7	0.9137	1.9479	1.5622	1.1244	1.1244	0.7801
2.5	80.7	0.5638	1.9217	1.5449	1.0095	1.0095	0.4990
3.0	86.0	0.3661	1.8169	1.4526	0.8716	0.8716	0.3317
3.5	90.4	0.2682	1.8823	1.5031	0.9291	0.9291	0.2593
4.0	93.5	0.2026	1.9301	1.5745	0.9216	0.9216	0.2129
			Type I Error Rates				
0.0	37.5	1.00	0.61	0.77	1.00	1.00	1.00
0.5	47.9	1.00	0.54	0.66	0.99	0.99	1.00
1.0	56.8	1.00	0.52	0.60	0.97	0.93	1.00
1.5	65.6	1.00	0.48	0.50	0.96	0.90	0.98
2.0	73.7	0.97	0.47	0.48	0.95	0.85	0.92
2.5	80.7	0.88	0.46	0.48	0.94	0.85	0.82
3.0	86.0	0.75	0.41	0.41	0.94	0.84	0.71
3.5	90.4	0.61	0.38	0.40	0.93	0.85	0.59
4.0	93.5	0.51	0.39	0.40	0.93	0.83	0.55

FIGURE 7
Distribution of All Estimators:
Panel RE Case, α=1, Publication Bias against Wrong Sign

Kernel density estimates
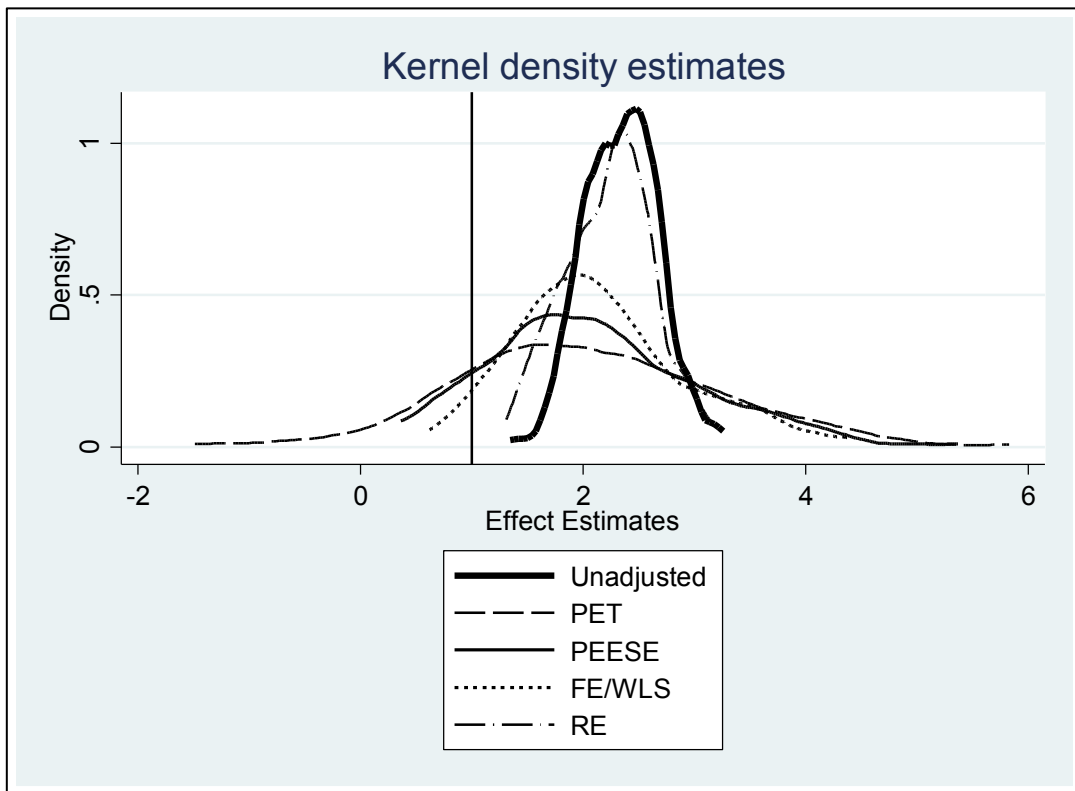
APPENDIX:
Stata .do files to produce all tables and figures in the paper

Note #1: These .do files are included to make it possible for readers to confirm this study's findings, and investigate alternative DGP specifications. Many of the programs perform 1000 simulations and take several hours to run. However, the running time can be decreased without substantially affecting the results by reducing the number of replications to 100.

Note #2: Some programs consist of two parts. Part A should be run first.

.do file for TABLE 1 – Part A

NOTE: **The yellow-highlighted part of the code determines which type of publication bias is operative**

```
clear
set more off
set seed 52256
program drop _all
program define sample, rclass
version 13.1

syntax, alpha(real) studies(integer) select(real) obs(integer)

// Remove existing variables
drop _all

//We first create the matrix to store the results of each study
matrix A = J(`studies',1,0)
matrix B = J(`studies',1,0)
matrix C = J(`studies',1,0)

        forvalues i = 1/`studies' {
        clear
        // STEP ONE: Create the data for each study and estimate an effect
        set obs `obs'
        generate x = rnormal()
        // Note that each "study" has the same number of observations (100)
        // but differ in the variance of their respective error terms.  This
        // causes the estimate of the effect to be estimated with varying degrees
        // of precision
        scalar lambda = 0.2+30*runiform()
        generate e = lambda*rnormal()
        generate y = 1 + `alpha'*x + e

        quiety regress y x
        scalar coef = _b[x]
        scalar secoef = _se[x]
        scalar tcoef = coef/secoef

        // First run this program once to get the pre-publication bias sample data
        // To get post-publication bias sample data, uncomment one of the two sections
        // below.

                /*if abs(tcoef) < 2 {
                scalar dummy = cond(runiform()<`select',1,.)
                scalar coef = dummy*coef
                scalar secoef = dummy*secoef
                scalar tcoef = dummy*tcoef
                 }
                */

                /*if coef < 0 {
                scalar dummy = cond(runiform()<`select',1,.)
                scalar coef = dummy*coef
```

```
                    scalar secoef = dummy*secoef
                    scalar tcoef = dummy*tcoef
                    }
                    */

          matrix A[`i',1] = coef
          matrix B[`i',1] = secoef
          matrix C[`i',1] = tcoef
          }


matrix bob = A,B,C
svmat bob
rename bob1 effect
rename bob2 seeffect
rename bob3 teffect
generate pet = (1/seeffect)

summ effect, detail
return scalar effectMED = r(p50)
return scalar effectMIN = r(min)
return scalar effectP5 = r(p5)
return scalar effectP95 = r(p95)
return scalar effectMAX = r(max)
return scalar N = r(N)

summ teffect, detail
return scalar teffectMED = r(p50)
return scalar teffectMIN = r(min)
return scalar teffectP5 = r(p5)
return scalar teffectP95 = r(p95)
return scalar teffectMAX = r(max)

summ pet, detail
return scalar petMED = r(p50)
return scalar petMIN = r(min)
return scalar petP5 = r(p5)
return scalar petP95 = r(p95)
return scalar petMAX = r(max)

end
```

.do file for TABLE 1 – Part B

```
// This program takes about 20 minutes to run on my laptop for each section
etime, start
drop _all
clear
graph drop _all
set more off
set seed 52256
matrix RESULTS = J(3,6,0)
local studies = 300
// Select is used to set the probability of being included in the data set
// when the study is subject to publication bias, either because (i) abs(tstat) < 2, or
// coef < 0.
local select = 0.10
local obs = 100
local alpha = 1

simulate effectMED = r(effectMED) effectMIN = r(effectMIN) effectP5 = r(effectP5) ///
effectP95 = r(effectP95) effectMAX = r(effectMAX)  ///
teffectMED = r(teffectMED) teffectMIN = r(teffectMIN) teffectP5 = r(teffectP5) ///
teffectP95 = r(teffectP95) teffectMAX = r(teffectMAX)  ///
petMED = r(petMED) petMIN = r(petMIN) petP5 = r(petP5) ///
petP95 = r(petP95) petMAX = r(petMAX)  N = r(N), ///
reps(1000): sample, alpha(`alpha') studies(`studies') select(`select') obs(`obs')

summ effectMED, meanonly
matrix RESULTS[1,1] = r(mean)
summ effectMIN, meanonly
matrix RESULTS[1,2] = r(mean)
summ effectP5, meanonly
matrix RESULTS[1,3] = r(mean)
summ effectP95, meanonly
matrix RESULTS[1,4] = r(mean)
summ effectMAX, meanonly
matrix RESULTS[1,5] = r(mean)
summ N, meanonly
matrix RESULTS[1,6] = r(mean)

summ teffectMED, meanonly
matrix RESULTS[2,1] = r(mean)
summ teffectMIN, meanonly
matrix RESULTS[2,2] = r(mean)
summ teffectP5, meanonly
matrix RESULTS[2,3] = r(mean)
summ teffectP95, meanonly
matrix RESULTS[2,4] = r(mean)
summ teffectMAX, meanonly
matrix RESULTS[2,5] = r(mean)
summ N, meanonly
matrix RESULTS[2,6] = r(mean)

summ petMED, meanonly
matrix RESULTS[3,1] = r(mean)
summ petMIN, meanonly
matrix RESULTS[3,2] = r(mean)
```

```
summ petP5, meanonly
matrix RESULTS[3,3] = r(mean)
summ petP95, meanonly
matrix RESULTS[3,4] = r(mean)
summ petMAX, meanonly
matrix RESULTS[3,5] = r(mean)
summ N, meanonly
matrix RESULTS[3,6] = r(mean)

matrix colnames RESULTS = MED MIN P5 P95 MAX N
matrix rownames RESULTS = EFFECT TEFFECT PET
matrix list RESULTS

etime
```

.do file for FIGURE 1

NOTE: **The yellow-highlighted part of the code determines which type of publication bias is operative**

```
clear
set seed 52256
set more off
local studies = 300
local alpha = 1
local select = 0.10
matrix A = J(`studies',1,0)
matrix B = J(`studies',1,0)
matrix C = J(`studies',1,0)

        forvalues i = 1/`studies' {
        // STEP ONE: Create the data for each study and estimate an effect
        clear
        set obs 100
        generate x = rnormal()
        scalar lambda = 0.2+30*runiform()
        // Note that each "study" has a difference error variance, causing the estimate
        // of the effect to be estimated with varying degrees of precision
        generate e = lambda*rnormal()
        generate y = 1 + `alpha'*x + e

        quietly regress y x
        scalar coef = _b[x]
        scalar secoef = _se[x]
        scalar tcoef = abs(coef/secoef)

        // First run this program once to get the pre-publication study sample data
        // To get post-publication study sample data, uncomment one of the two sections
        // below.

                /*if tcoef < 2 {
                scalar dummy = cond(runiform()<`select',1,.)
                scalar coef = dummy*coef
                scalar secoef = dummy*secoef
                scalar tcoef = dummy*tcoef
                }
                */

                /*if coef < 0 {
                scalar dummy = cond(runiform()<`select',1,.)
                scalar coef = dummy*coef
                scalar secoef = dummy*secoef
                scalar tcoef = dummy*tcoef
                }
                */

        matrix A[`i',1] = coef
        matrix B[`i',1] = secoef
        matrix C[`i',1] = tcoef
        }
```

```
matrix bob = A,B,C
svmat bob
rename bob1 effect
rename bob2 seeffect
rename bob3 teffect
generate pet = (1/seeffect)

metafunnel effect seeffect, xline(`alpha') ylabel(#8) nolines
```

.do file for TABLES 2 and 3 – Part A

NOTE: **The yellow-highlighted part of the code determines which type of publication bias is operative (TABLE 2 = bias against insignificance; TABLE 3 = bias against wrong sign)**

```
set more off
program drop _all
program define FEpbias, rclass
version 13.1

syntax, alpha(real) studies(integer) select(real) obs(integer)

// Remove existing variables
drop _all

//We first create the matrix to store the results of each study
matrix A = J(`studies',1,0)
matrix B = J(`studies',1,0)
matrix C = J(`studies',1,0)

// Create the data for each study and estimate an effect

        forvalues i = 1/`studies' {
        clear
        set obs `obs'
        generate x = rnormal()
        scalar lambda = 0.2 + 30*runiform()
        // Note that each "study" has the same number of observations (100)
        // but differ in the variance of their respective error terms.  This
        // causes the estimate of the effect to be estimated with varying degrees
        // of precision
        generate e = lambda*rnormal()
        generate y = 1 + `alpha'*x + e

        quietly regress y x
        scalar coef = _b[x]
        scalar secoef = _se[x]
        scalar tcoef = coef/secoef

                if abs(tcoef) < 2 {

                // An alternative criterion is that results that show a negative effect
                // have a harder time getting published.  To study that case, substitute
                // the line below for the line above

                // if coef < 0 {

                scalar dummy = cond(runiform()<`select',1,.)
                // The statement above creates a dummy variable that randomly selects which
                // studies will get "published" if they fail to meet the "publication criterion"
                // either (i) abs(tcoef) >= 2 or (ii) if coef >= 0.  Studies that are not "published"
                // receive missing values and thus are not included in the "meta-analysis."
                scalar coef = dummy*coef
                scalar secoef = dummy*secoef
```

```
                    scalar tcoef = dummy*tcoef
                    }


            matrix A[`i',1] = coef
            matrix B[`i',1] = secoef
            matrix C[`i',1] = tcoef
            }


// The next set of commands moves the data out of matrices and reformats them as
// standard Stata data series.  We have now completed generating our individual
// studies and we now move into the "meta-analysis" stage.
matrix bob = A,B,C
svmat bob
rename bob1 effect
rename bob2 seeffect
rename bob3 teffect
generate pet = (1/seeffect)

// This estimate produces the OLS estimate of the effect
regress effect
return scalar effect_OLS = _b[_cons]
test _b[_cons] = `alpha'
return scalar pvalue_OLS = r(p)

// This estimate produces the PET version of the "publication bias"-
// corrected effect estimate
regress teffect pet, vce(robust)
return scalar effect_PET = _b[pet]
test pet = `alpha'
return scalar pvalue_PET = r(p)

// This estimate produces the PEESE version of the "publication bias"-
// corrected effect estimate
regress teffect seeffect pet, noc vce(robust)
return scalar effect_PEESE = _b[pet]
test pet = `alpha'
return scalar pvalue_PEESE = r(p)

// This estimate produces the FE estimate of the effect
generate constant = 1
vwls effect constant, sd(seeffect) nocon
return scalar effect_FE = _b[constant]
test _b[constant] = `alpha'
return scalar pvalue_FE = r(p)

// This estimate produces the WLS estimate of the effect
regress teffect pet, noc
return scalar effect_WLS = _b[pet]
test pet = `alpha'
return scalar pvalue_WLS = r(p)
```

54

```
// This estimate produces the RE estimate of the effect
// NOTE: We use the Method of Moments (mm) option of metareg
// because the maximum likelihood procedure had too many instances
// of failure to optimize.  Method of Moments does not require
// iteration and thus avoids this problem.
metareg effect , wsse(seeffect) mm
matrix bill = e(b)
return scalar effect_RE = bill[1,1]
test _cons = `alpha'
return scalar pvalue_RE = r(p)

// This last command keeps track of how many studies are in our "meta-analysis"
return scalar N = e(N)

end
```

.do file for TABLES 2 and 3 – Part B

```
// This program takes about 3 hours to run on my computer
etime, start
drop _all
clear
graph drop _all
set more off
set seed 52256
matrix EFFECT = J(9,7,0)
matrix MSE = J(9,7,0)
matrix TEST = J(9,7,0)
local studies = 300

// Select is used to set the probability of being included in the data set
// when the study is subject to publication bias, either because (i) abs(tstat) < 2, or
// coef < 0.
local select = 0.10
local obs = 100

        local i = 1
        foreach alpha in 0 0.5 1 1.5 2 2.5 3 3.5 4 {
        simulate effect_OLS = r(effect_OLS) effect_FE = r(effect_FE) effect_RE = r(effect_RE) ///
        effect_PET = r(effect_PET) effect_PEESE = r(effect_PEESE)  ///
        pvalue_OLS = r(pvalue_OLS) pvalue_FE = r(pvalue_FE) pvalue_RE = r(pvalue_RE) ///
        pvalue_PET = r(pvalue_PET) pvalue_PEESE = r(pvalue_PEESE) N = r(N) ///
        effect_WLS = r(effect_WLS) pvalue_WLS = r(pvalue_WLS) , ///
        reps(1000): FEpbias, alpha(`alpha') studies(`studies') select(`select') obs(`obs')

        summ N, meanonly
        matrix EFFECT[`i',1] = r(mean)
        summ effect_OLS, meanonly
        matrix EFFECT[`i',2] = r(mean)
        summ effect_PET, meanonly
        matrix EFFECT[`i',3] = r(mean)
        summ effect_PEESE, meanonly
        matrix EFFECT[`i',4] = r(mean)
        summ effect_FE, meanonly
        matrix EFFECT[`i',5] = r(mean)
        summ effect_WLS, meanonly
         matrix EFFECT[`i',6] = r(mean)
         summ effect_RE, meanonly
         matrix EFFECT[`i',7] = r(mean)

        summ N, meanonly
        matrix MSE[`i',1] = r(mean)
        generate mse_OLS = (effect_OLS - `alpha')^2
        summ mse_OLS, meanonly
        matrix MSE[`i',2] = r(mean)
        generate mse_PET = (effect_PET - `alpha')^2
        summ mse_PET, meanonly
        matrix MSE[`i',3] = r(mean)
        generate mse_PEESE = (effect_PEESE - `alpha')^2
        summ mse_PEESE, meanonly
        matrix MSE[`i',4] = r(mean)
        generate mse_FE = (effect_FE - `alpha')^2
```

56

```
        summ mse_FE, meanonly
        matrix MSE[`i',5] = r(mean)
        generate mse_WLS = (effect_WLS - `alpha')^2
        summ mse_WLS, meanonly
        matrix MSE[`i',6] = r(mean)
        generate mse_RE = (effect_RE - `alpha')^2
        summ mse_RE, meanonly
        matrix MSE[`i',7] = r(mean)

        summ N, meanonly
        matrix TEST[`i',1] = r(mean)
        generate RROLS = 0
        replace RROLS = cond(pvalue_OLS<0.05,1,0)
        summ RROLS, meanonly
        matrix TEST[`i',2] = r(mean)
        generate RRPET = 0
        replace RRPET = cond(pvalue_PET<0.05,1,0)
        summ RRPET, meanonly
        matrix TEST[`i',3] = r(mean)
        generate RRPEESE = 0
        replace RRPEESE = cond(pvalue_PEESE<0.05,1,0)
        summ RRPEESE, meanonly
        matrix TEST[`i',4] = r(mean)
        generate RRFE = 0
        replace RRFE = cond(pvalue_FE<0.05,1,0)
        summ RRFE, meanonly
        matrix TEST[`i',5] = r(mean)
        generate RRWLS = 0
        replace RRWLS = cond(pvalue_WLS<0.05,1,0)
        summ RRWLS, meanonly
        matrix TEST[`i',6] = r(mean)
        generate RRRE = 0
        replace RRRE = cond(pvalue_RE<0.05,1,0)
        summ RRRE, meanonly
        matrix TEST[`i',7] = r(mean)

        local `++i'
        }

matrix colnames EFFECT = N OLS PET PEESE FE WLS RE
matrix rownames EFFECT = A0 A0P5 A1 A1P5 A2 A2P5 A3 A3P5 A4
matrix colnames MSE = N OLS PET PEESE FE WLS RE
matrix rownames MSE = A0 A0P5 A1 A1P5 A2 A2P5 A3 A3P5 A4
matrix colnames TEST = N OLS PET PEESE FE WLS RE
matrix rownames TEST = A0 A0P5 A1 A1P5 A2 A2P5 A3 A3P5 A4
matrix list EFFECT
matrix list MSE
matrix list TEST

etime
```

.do file for TABLE 4 – Part A

NOTE: **The yellow-highlighted part of the code determines which type of publication bias is operative**

```
clear
set more off
program drop _all
program define sample, rclass
version 13.1

syntax, alpha(real) studies(integer) select(real) obs(integer)

// Remove existing variables
drop _all

//We first create the matrix to store the results of each study
matrix A = J(`studies',1,0)
matrix B = J(`studies',1,0)
matrix C = J(`studies',1,0)

        forvalues i = 1/`studies' {
        clear
        // STEP ONE: Create the data for each study and estimate an effect
        set obs `obs'
        generate x = rnormal()
        // Note that each "study" has the same number of observations (100)
        // but differ in the variance of their respective error terms.  This
        // causes the estimate of the effect to be estimated with varying degrees
        // of precision
        scalar lambda = 0.5+30*runiform()
        generate e = lambda*rnormal()
        scalar alpha = `alpha' + rnormal()
        generate y = 1 + alpha*x + e

        quietly regress y x
        scalar coef = _b[x]
        scalar secoef = _se[x]
        scalar tcoef = coef/secoef

        // First run this program once to get the pre-publication study sample data
        // To get post-publication study sample data, uncomment one of the two sections
        // below.

                /*if abs(tcoef) < 2 {
                scalar dummy = cond(runiform()<`select',1,.)
                scalar coef = dummy*coef
                scalar secoef = dummy*secoef
                scalar tcoef = dummy*tcoef
                }
                */

                /*if coef < 0 {
                scalar dummy = cond(runiform()<`select',1,.)
                scalar coef = dummy*coef
```

```
                    scalar secoef = dummy*secoef
                    scalar tcoef = dummy*tcoef
                    }
                    */

        matrix A[`i',1] = coef
        matrix B[`i',1] = secoef
        matrix C[`i',1] = tcoef
        }

matrix bob = A,B,C
svmat bob
rename bob1 effect
rename bob2 seeffect
rename bob3 teffect
generate pet = (1/seeffect)

metareg effect , wsse(seeffect) mm
return scalar I2 = e(I2)

summ effect, detail
return scalar effectMED = r(p50)
return scalar effectMIN = r(min)
return scalar effectP5 = r(p5)
return scalar effectP95 = r(p95)
return scalar effectMAX = r(max)
return scalar N = r(N)

summ teffect, detail
return scalar teffectMED = r(p50)
return scalar teffectMIN = r(min)
return scalar teffectP5 = r(p5)
return scalar teffectP95 = r(p95)
return scalar teffectMAX = r(max)

summ pet, detail
return scalar petMED = r(p50)
return scalar petMIN = r(min)
return scalar petP5 = r(p5)
return scalar petP95 = r(p95)
return scalar petMAX = r(max)

end
```

.do file for TABLE 4 – Part B

```
// This program takes about 20 minutes to run on my laptop for each section
etime, start
drop _all
clear
graph drop _all
set more off
set seed 52256
matrix RESULTS = J(4,6,0)
local studies = 300
// Select is used to set the probability of being included in the data set
// when the study is subject to publication bias, either because (i) abs(tstat) < 2, or
// coef < 0.
local select = 0.10
local obs = 100
local alpha = 1

simulate effectMED = r(effectMED) effectMIN = r(effectMIN) effectP5 = r(effectP5) ///
effectP95 = r(effectP95) effectMAX = r(effectMAX)  ///
teffectMED = r(teffectMED) teffectMIN = r(teffectMIN) teffectP5 = r(teffectP5) ///
teffectP95 = r(teffectP95) teffectMAX = r(teffectMAX)  ///
petMED = r(petMED) petMIN = r(petMIN) petP5 = r(petP5) ///
petP95 = r(petP95) petMAX = r(petMAX)  N = r(N) I2 = r(I2), ///
reps(1000): sample, alpha(`alpha') studies(`studies') select(`select') obs(`obs')

summ effectMED, meanonly
matrix RESULTS[1,1] = r(mean)
summ effectMIN, meanonly
matrix RESULTS[1,2] = r(mean)
summ effectP5, meanonly
matrix RESULTS[1,3] = r(mean)
summ effectP95, meanonly
matrix RESULTS[1,4] = r(mean)
summ effectMAX, meanonly
matrix RESULTS[1,5] = r(mean)
summ N, meanonly
matrix RESULTS[1,6] = r(mean)

summ teffectMED, meanonly
matrix RESULTS[2,1] = r(mean)
summ teffectMIN, meanonly
matrix RESULTS[2,2] = r(mean)
summ teffectP5, meanonly
matrix RESULTS[2,3] = r(mean)
summ teffectP95, meanonly
matrix RESULTS[2,4] = r(mean)
summ teffectMAX, meanonly
matrix RESULTS[2,5] = r(mean)
summ N, meanonly
matrix RESULTS[2,6] = r(mean)

summ petMED, meanonly
matrix RESULTS[3,1] = r(mean)
summ petMIN, meanonly
matrix RESULTS[3,2] = r(mean)
```

```
summ petP5, meanonly
matrix RESULTS[3,3] = r(mean)
summ petP95, meanonly
matrix RESULTS[3,4] = r(mean)
summ petMAX, meanonly
matrix RESULTS[3,5] = r(mean)
summ N, meanonly
matrix RESULTS[3,6] = r(mean)

summ I2, detail
matrix RESULTS[4,1] = r(mean)
matrix RESULTS[4,2] = r(min)
matrix RESULTS[4,3] = r(p5)
matrix RESULTS[4,4] = r(p95)
matrix RESULTS[4,5] = r(max)

summ N, meanonly
matrix RESULTS[4,6] = r(mean)

matrix colnames RESULTS = MED MIN P5 P95 MAX N
matrix rownames RESULTS = EFFECT TEFFECT PET HETERO
matrix list RESULTS

etime
```

.do file for FIGURE 2

NOTE: **The yellow-highlighted part of the code determines which type of publication bias is operative**

```
clear
set seed 52256
set more off
local studies = 300
local alpha = 1
local select = 0.10
matrix A = J(`studies',1,0)
matrix B = J(`studies',1,0)
matrix C = J(`studies',1,0)

        forvalues i = 1/`studies' {
        // STEP ONE: Create the data for each study and estimate an effect
        clear
        set obs 100
        generate x = rnormal()
        // Note that each "study" has a difference error variance, causing the estimate
        // of the effect to be estimated with varying degrees of precision
        scalar lambda = 0.5+30*runiform()
        generate e = lambda*rnormal()
        scalar alpha = `alpha' + rnormal()
        generate y = 1 + alpha*x + e

        quietly regress y x
        scalar coef = _b[x]
        scalar secoef = _se[x]
        scalar tcoef = abs(coef/secoef)

        // First run this program once to get the pre-publication study sample data
        // To get post-publication study sample data, uncomment one of the two sections
        // below.

                /*if tcoef < 2 {
                scalar dummy = cond(runiform()<`select',1,.)
                scalar coef = dummy*coef
                scalar secoef = dummy*secoef
                scalar tcoef = dummy*tcoef
                }
                */

                /*if coef < 0 {
                scalar dummy = cond(runiform()<`select',1,.)
                scalar coef = dummy*coef
                scalar secoef = dummy*secoef
                scalar tcoef = dummy*tcoef
                }
                */
```

```
        matrix A[`i',1] = coef
        matrix B[`i',1] = secoef
        matrix C[`i',1] = tcoef
        }

matrix bob = A,B,C
svmat bob
rename bob1 effect
rename bob2 seeffect
rename bob3 teffect
generate pet = (1/seeffect)

metafunnel effect seeffect, xline(`alpha') ylabel(#8) nolines
```

.do file for TABLES 5 and 6 – Part A

NOTE: **The yellow-highlighted part of the code determines which type of publication bias is operative (TABLE 5 = bias against insignificance; TABLE 6 = bias against wrong sign)**

```
set more off
program drop _all
program define REpbias, rclass
version 13.1

syntax, alpha(real) studies(integer) select(real) obs(integer)

// Remove existing variables
drop _all

//We first create the matrix to store the results of each study
matrix A = J(`studies',1,0)
matrix B = J(`studies',1,0)
matrix C = J(`studies',1,0)

        forvalues i = 1/`studies' {
        clear
        // STEP ONE: Create the data for each study and estimate an effect
        set obs `obs'
        generate x = rnormal()
        // Note that each "study" has the same number of observations (100)
        // but differ in the variance of their respective error terms.  This
        // causes the estimate of the effect to be estimated with varying degrees
        // of precision
        scalar lambdai = 0.5+30*runiform()
        generate e = lambdai*rnormal()
        scalar alphai = `alpha' + rnormal()
        generate y = 1 + alphai*x + e

        quietly regress y x
        scalar coef = _b[x]
        scalar secoef = _se[x]
        scalar tcoef = coef/secoef

                //if abs(tcoef) < 2 {
                // An alternative criterion is that results that show a negative effect
                // have a harder time getting published.  To study that case, substitute
                // the line below for the line above
                //if coef < 0 {
                scalar dummy = cond(runiform()<`select',1,.)
                // The statement above creates a dummy variable that randomly selects which
                // studies will get "published" if they fail to meet the "publication criterion"
                // either (i) abs(tcoef) >= 2 or (ii) if coef >= 0.  Studies that are not "published"
                // receive missing values and thus are not included in the "meta-analysis."
                scalar coef = dummy*coef
                scalar secoef = dummy*secoef
                scalar tcoef = dummy*tcoef
                }
```

```
        matrix A[`i',1] = coef
        matrix B[`i',1] = secoef
        matrix C[`i',1] = tcoef
        }

// The next set of commands moves the data out of matrices and reformats them as
// standard Stata data series.  We have now completed generating our individual
// studies and we now move into the "meta-analysis" stage.
matrix bob = A,B,C
svmat bob
rename bob1 effect
rename bob2 seeffect
rename bob3 teffect
generate pet = (1/seeffect)

// This estimate produces the OLS estimate of the effect
regress effect
return scalar effect_OLS = _b[_cons]
test _b[_cons] = `alpha'
return scalar pvalue_OLS = r(p)

// This estimate produces the PET version of the "publication bias"-
// corrected effect estimate
regress teffect pet, vce(robust)
return scalar effect_PET = _b[pet]
test pet = `alpha'
return scalar pvalue_PET = r(p)

// This estimate produces the PEESE version of the "publication bias"-
// corrected effect estimate
regress teffect seeffect pet, noc vce(robust)
return scalar effect_PEESE = _b[pet]
test pet = `alpha'
return scalar pvalue_PEESE = r(p)

// This estimate produces the FE estimate of the effect
generate constant = 1
vwls effect constant, sd(seeffect) nocon
return scalar effect_FE = _b[constant]
test _b[constant] = `alpha'
return scalar pvalue_FE = r(p)

// This estimate produces the WLS estimate of the effect
regress teffect pet, noc
return scalar effect_WLS = _b[pet]
test pet = `alpha'
return scalar pvalue_WLS = r(p)
```

```
// This estimate produces the RE estimate of the effect
// NOTE: We use the Method of Moments (mm) option of metareg
// because the maximum likelihood procedure had too many instances
// of failure to optimize.  Method of Moments does not require
// iteration and thus avoids this problem.
metareg effect , wsse(seeffect) mm
matrix bill = e(b)
return scalar effect_RE = bill[1,1]
test _cons = `alpha'
return scalar pvalue_RE = r(p)

// This last command keeps track of how many studies are in our "meta-analysis"
return scalar N = e(N)

end
```

.do file for TABLES 5 and 6 – Part B

```
// This program takes about 3 hours to run on my computer
etime, start
drop _all
clear
graph drop _all
set more off
set seed 52256
matrix EFFECT = J(9,7,0)
matrix MSE = J(9,7,0)
matrix TEST = J(9,7,0)
local studies = 300
// Select is used to set the probability of being included in the data set
// when the study is subject to publication bias, either because (i) abs(tstat) < 2, or
// coef < 0.
local select = 0.10
local obs = 100

local i = 1

        foreach alpha in 0 0.5 1 1.5 2 2.5 3 3.5 4 {
        simulate effect_OLS = r(effect_OLS) effect_FE = r(effect_FE) effect_RE = r(effect_RE) ///
        effect_PET = r(effect_PET) effect_PEESE = r(effect_PEESE)  ///
        pvalue_OLS = r(pvalue_OLS) pvalue_FE = r(pvalue_FE) pvalue_RE = r(pvalue_RE) ///
        pvalue_PET = r(pvalue_PET) pvalue_PEESE = r(pvalue_PEESE) N = r(N) ///
        effect_WLS = r(effect_WLS) pvalue_WLS = r(pvalue_WLS) , ///
        reps(1000): REpbias, alpha(`alpha') studies(`studies') select(`select') obs(`obs')

        summ N, meanonly
        matrix EFFECT[`i',1] = r(mean)
        summ effect_OLS, meanonly
        matrix EFFECT[`i',2] = r(mean)
        summ effect_PET, meanonly
        matrix EFFECT[`i',3] = r(mean)
        summ effect_PEESE, meanonly
        matrix EFFECT[`i',4] = r(mean)
        summ effect_FE, meanonly
        matrix EFFECT[`i',5] = r(mean)
        summ effect_WLS, meanonly
        matrix EFFECT[`i',6] = r(mean)
        summ effect_RE, meanonly
        matrix EFFECT[`i',7] = r(mean)

        summ N, meanonly
        matrix MSE[`i',1] = r(mean)
        generate mse_OLS = (effect_OLS - `alpha')^2
        summ mse_OLS, meanonly
        matrix MSE[`i',2] = r(mean)
        generate mse_PET = (effect_PET - `alpha')^2
        summ mse_PET, meanonly
        matrix MSE[`i',3] = r(mean)
        generate mse_PEESE = (effect_PEESE - `alpha')^2
        summ mse_PEESE, meanonly
        matrix MSE[`i',4] = r(mean)
        generate mse_FE = (effect_FE - `alpha')^2
```

```
        summ mse_FE, meanonly
        matrix MSE[`i',5] = r(mean)
        generate mse_WLS = (effect_WLS - `alpha')^2
        summ mse_WLS, meanonly
        matrix MSE[`i',6] = r(mean)
        generate mse_RE = (effect_RE - `alpha')^2
        summ mse_RE, meanonly
        matrix MSE[`i',7] = r(mean)

        summ N, meanonly
        matrix TEST[`i',1] = r(mean)
        generate RROLS = 0
        replace RROLS = cond(pvalue_OLS<0.05,1,0)
        summ RROLS, meanonly
        matrix TEST[`i',2] = r(mean)
        generate RRPET = 0
        replace RRPET = cond(pvalue_PET<0.05,1,0)
        summ RRPET, meanonly
        matrix TEST[`i',3] = r(mean)
        generate RRPEESE = 0
        replace RRPEESE = cond(pvalue_PEESE<0.05,1,0)
        summ RRPEESE, meanonly
        matrix TEST[`i',4] = r(mean)
        generate RRFE = 0
        replace RRFE = cond(pvalue_FE<0.05,1,0)
        summ RRFE, meanonly
        matrix TEST[`i',5] = r(mean)
        generate RRWLS = 0
        replace RRWLS = cond(pvalue_WLS<0.05,1,0)
        summ RRWLS, meanonly
        matrix TEST[`i',6] = r(mean)
        generate RRRE = 0
        replace RRRE = cond(pvalue_RE<0.05,1,0)
        summ RRRE, meanonly
        matrix TEST[`i',7] = r(mean)

        local `++i'
        }

matrix colnames EFFECT = N OLS PET PEESE FE WLS RE
matrix rownames EFFECT = A0 A0P5 A1 A1P5 A2 A2P5 A3 A3P5 A4
matrix colnames MSE = N OLS PET PEESE FE WLS RE
matrix rownames MSE = A0 A0P5 A1 A1P5 A2 A2P5 A3 A3P5 A4
matrix colnames TEST = N OLS PET PEESE FE WLS RE
matrix rownames TEST = A0 A0P5 A1 A1P5 A2 A2P5 A3 A3P5 A4
matrix list EFFECT
matrix list MSE
matrix list TEST

etime
```

.do file for FIGURE 3 – Part A

NOTE: **The yellow-highlighted part of the code determines which type of publication bias is operative**


```
set more off
program drop _all
program define REfig, rclass
version 13.1

syntax, alpha(real) studies(integer) select(real) obs(integer)
// Remove existing variables
drop _all

//We first create the matrix to store the results of each study
matrix A = J(`studies',1,0)
matrix B = J(`studies',1,0)
matrix C = J(`studies',1,0)

        forvalues i = 1/`studies' {
        clear
        // STEP ONE: Create the data for each study and estimate an effect
        set obs `obs'
        generate x = rnormal()
        // Note that each "study" has the same number of observations (100)
        // but differ in the variance of their respective error terms.  This
        // causes the estimate of the effect to be estimated with varying degrees
        // of precision
        scalar lambda = 0.5+30*runiform()
        generate e = lambda*rnormal()
        scalar alpha = `alpha' + rnormal()
        generate y = 1 + alpha*x + e

        regress y x
        scalar coef = _b[x]
        scalar secoef = _se[x]
        scalar tcoef = coef/secoef

                // if abs(tcoef) < 2 {
                // An alternative criterion is that results that show a negative effect
                // have a harder time getting published.  To study that case, substitute
                // the line below for the line above
                if coef < 0 {
                scalar dummy = cond(runiform()<`select',1,.)
                // The statement above creates a dummy variable that randomly selects which
                // studies will get "published" if they fail to meet the "publication criterion"
                // either (i) abs(tcoef) >= 2 or (ii) if coef >= 0.  Studies that are not "published"
                // receive missing values and thus are not included in the "meta-analysis."
                scalar coef = dummy*coef
                scalar secoef = dummy*secoef
                scalar tcoef = dummy*tcoef
                }
```

```
        matrix A[`i',1] = coef
        matrix B[`i',1] = secoef
        matrix C[`i',1] = tcoef
        }

// The next set of commands moves the data out of matrices and reformats them as
// standard Stata data series.  We have now completed generating our individual
// studies and we now move into the "meta-analysis" stage.
matrix bob = A,B,C
svmat bob
rename bob1 effect
rename bob2 seeffect
rename bob3 teffect
generate pet = (1/seeffect)

// This estimate produces the RE estimate of the effect
metareg effect , wsse(seeffect) mm
matrix bill = e(b)
return scalar effect_RE = bill[1,1]

// This estimate produces the PEESE version of the "publication bias"-
// corrected effect estimate.
regress teffect seeffect pet, noc
return scalar effect_PEESE = _b[pet]

end
```

.do file for FIGURE 3 – Part B

NOTE: The yellow-highlighted part of the code determines which type of publication bias is operative

```
// This program takes about 3 minutes to run on my laptop
etime, start
drop _all
clear
graph drop _all
set more off
set seed 52256
matrix EFFECT = J(9,6,0)
matrix MSE = J(9,6,0)
matrix TEST = J(9,6,0)
local studies = 300
// Select is used to set the probability of being included in the data set
// when the study is subject to publication bias, either because (i) abs(tstat) < 2, or
// coef < 0.
local select = 0.10
local obs = 100

local i = 1

        foreach alpha in 3  {
        simulate effect_RE = r(effect_RE) effect_PEESE = r(effect_PEESE), ///
        reps(100): REfig, studies(`studies') alpha(`alpha') select(`select') obs(`obs')

        local `++i'
        }

kdensity effect_PEESE  , xline(4) name(PEESE) //xlabel(-4(1) 4)
kdensity effect_RE , xline(4) name(Combined) addplot(kdensity effect_PEESE) //xlabel(-4(1) 4)

etime
```

.do file for TABLE 7 – Part A

NOTE: **The yellow-highlighted part of the code determines which type of publication bias is operative**

```
set more off
program drop _all
program define PANELsample, rclass
version 13.1

syntax, studies(integer) estperstudy(integer) totalobs(integer) alpha(real) ///
theta(real) obs(integer)

// Remove existing variables
drop _all

//We first create the matrix to store the results of each study
set matsize 10000
matrix A = J(`totalobs',1,.)
matrix B = J(`totalobs',1,.)
matrix C = J(`totalobs',1,.)
matrix D = J(`totalobs',1,.)

        forvalues i = 1/`studies' {
        scalar lambdai = 0.5+30*runiform()
        scalar alphai = `alpha'+2*rnormal()

                forvalues j = 1/`estperstudy' {
                // STEP ONE: Create the data for each study and estimate an effect
                clear
                set obs 100
                generate x = rnormal()
                // Note that each "study" has a difference error variance, causing the estimate
                // of the effect to be estimated with varying degrees of precision
                // This is for random effects
                scalar lambdaij = lambdai+`theta'*runiform()
                scalar alphaij = alphai+0.5*rnormal()
                generate e = lambdaij*rnormal()
                generate y = 1 + alphaij*x + e

                quietly regress y x
                scalar coef = _b[x]
                scalar secoef = _se[x]
                scalar tcoef = coef/secoef
                scalar ID = `i'
                scalar obsno = (`i'-1)*`estperstudy'+`j'

                matrix A[obsno,1] = coef
                matrix B[obsno,1] = secoef
                matrix C[obsno,1] = tcoef
                matrix D[obsno,1] = ID
                }
        }
```

```
matrix bob = A,B,C,D
svmat bob
rename bob1 effect
rename bob2 seeffect
rename bob3 teffect
rename bob4 ID
generate pet = (1/seeffect)

// First run this program once to get the pre-publication study sample data
// To get post-publication study sample data, uncomment one of the two sections
// below.
generate dummy = 1
//replace dummy = 0 if abs(teffect) < 2
//replace dummy = 0 if effect < 0
by ID, sort: egen select = mean(dummy)
replace effect = cond(select<0.65,.,effect)
replace seeffect = cond(select<0.65,.,seeffect)
replace teffect = cond(select<0.65,.,teffect)
replace pet = cond(select<0.65,.,pet)

metareg effect , wsse(seeffect) mm
return scalar I2 = e(I2)

summ effect, detail
return scalar effectMED = r(p50)
return scalar effectMIN = r(min)
return scalar effectP5 = r(p5)
return scalar effectP95 = r(p95)
return scalar effectMAX = r(max)
return scalar N = r(N)

summ teffect, detail
return scalar teffectMED = r(p50)
return scalar teffectMIN = r(min)
return scalar teffectP5 = r(p5)
return scalar teffectP95 = r(p95)
return scalar teffectMAX = r(max)

summ pet, detail
return scalar petMED = r(p50)
return scalar petMIN = r(min)
return scalar petP5 = r(p5)
return scalar petP95 = r(p95)
return scalar petMAX = r(max)

end
```

```
// This program takes approximately 35 minutes to run on my computer
etime, start
drop _all
clear
graph drop _all
set more off
set seed 52256
matrix RESULTS = J(4,6,0)
local studies = 30
local estperstudy = 10
local totalobs = `studies'*`estperstudy'
local alpha = 1
local theta = 1
local obs = 100

simulate effectMED = r(effectMED) effectMIN = r(effectMIN) effectP5 = r(effectP5) ///
effectP95 = r(effectP95) effectMAX = r(effectMAX)  ///
teffectMED = r(teffectMED) teffectMIN = r(teffectMIN) teffectP5 = r(teffectP5) ///
teffectP95 = r(teffectP95) teffectMAX = r(teffectMAX)  ///
petMED = r(petMED) petMIN = r(petMIN) petP5 = r(petP5) ///
petP95 = r(petP95) petMAX = r(petMAX)  N = r(N) I2 = r(I2), ///
reps(1000): PANELsample, studies(`studies') estperstudy(`estperstudy') totalobs(`totalobs') alpha(`alpha') ///
theta(`theta') obs(`obs')

summ effectMED, meanonly
matrix RESULTS[1,1] = r(mean)
summ effectMIN, meanonly
matrix RESULTS[1,2] = r(mean)
summ effectP5, meanonly
matrix RESULTS[1,3] = r(mean)
summ effectP95, meanonly
matrix RESULTS[1,4] = r(mean)
summ effectMAX, meanonly
matrix RESULTS[1,5] = r(mean)
summ N, meanonly
matrix RESULTS[1,6] = r(mean)

summ teffectMED, meanonly
matrix RESULTS[2,1] = r(mean)
summ teffectMIN, meanonly
matrix RESULTS[2,2] = r(mean)
summ teffectP5, meanonly
matrix RESULTS[2,3] = r(mean)
summ teffectP95, meanonly
matrix RESULTS[2,4] = r(mean)
summ teffectMAX, meanonly
matrix RESULTS[2,5] = r(mean)
summ N, meanonly
matrix RESULTS[2,6] = r(mean)

summ petMED, meanonly
matrix RESULTS[3,1] = r(mean)
summ petMIN, meanonly
matrix RESULTS[3,2] = r(mean)
```

```
summ petP5, meanonly
matrix RESULTS[3,3] = r(mean)
summ petP95, meanonly
matrix RESULTS[3,4] = r(mean)
summ petMAX, meanonly
matrix RESULTS[3,5] = r(mean)
summ N, meanonly
matrix RESULTS[3,6] = r(mean)

summ I2, detail
matrix RESULTS[4,1] = r(mean)
matrix RESULTS[4,2] = r(min)
matrix RESULTS[4,3] = r(p5)
matrix RESULTS[4,4] = r(p95)
matrix RESULTS[4,5] = r(max)
summ N, meanonly
matrix RESULTS[4,6] = r(mean)

matrix colnames RESULTS = MED MIN P5 P95 MAX N
matrix rownames RESULTS = EFFECT TEFFECT PET HETERO
matrix list RESULTS

etime
```

.do file for FIGURE 6

NOTE: The yellow-highlighted part of the code determines which type of publication bias is operative

```
// This program takes about 5 seconds to run for each section
etime, start
clear
set seed 52256
set more off
local studies = 30
local estperstudy = 10
local totalobs = `studies'*`estperstudy'
local alpha = 1
local theta = 1
set matsize 10000
matrix A = J(`totalobs',1,.)
matrix B = J(`totalobs',1,.)
matrix C = J(`totalobs',1,.)
matrix D = J(`totalobs',1,.)

        forvalues i = 1/`studies' {
        scalar lambdai = 0.5+30*runiform()
        scalar alphai = `alpha'+2*rnormal()

                forvalues j = 1/`estperstudy' {
                // STEP ONE: Create the data for each study and estimate an effect
                clear
                set obs 100
                generate x = rnormal()
                // Note that each "study" has a difference error variance, causing the estimate
                // of the effect to be estimated with varying degrees of precision
                // This is for random effects
                scalar lambdai = lambdai+`theta'*runiform()
                scalar alphaij = alphai+0.5*rnormal()
                generate e = lambdai*rnormal()
                generate y = 1 + alphaij*x + e

                quietly regress y x
                scalar coef = _b[x]
                scalar secoef = _se[x]
                scalar tcoef = coef/secoef
                scalar ID = `i'
                scalar obsno = (`i'-1)*`estperstudy'+`j'

                matrix A[obsno,1] = coef
                matrix B[obsno,1] = secoef
                matrix C[obsno,1] = tcoef
                matrix D[obsno,1] = ID
                }

        }
```

```
matrix bob = A,B,C,D
svmat bob
rename bob1 effect
rename bob2 seeffect
rename bob3 teffect
rename bob4 ID
generate pet = (1/seeffect)

// First run this program once to get the pre-publication study sample data
// To get post-publication study sample data, uncomment one of the two sections
// below.
generate dummy = 1
//replace dummy = 0 if abs(teffect) < 2
//replace dummy = 0 if effect < 0
by ID, sort: egen select = mean(dummy)
replace effect = cond(select<0.65,.,effect)
replace seeffect = cond(select<0.65,.,seeffect)
replace teffect = cond(select<0.65,.,teffect)
replace pet = cond(select<0.65,.,pet)

metafunnel effect seeffect, xline(1) nolines ylabel(#8)

etime
```

.do file for TABLES 8 and 9 – Part A

<u>NOTE</u>: The yellow-highlighted part of the code determines which type of publication bias is operative (TABLE 8 = bias against insignificance; TABLE 9 = bias against wrong sign)

```
set more off
program drop _all
program define PANELpbias, rclass
version 13.1

syntax, studies(integer) estperstudy(integer) totalobs(integer) alpha(real) ///
theta(real) obs(integer)

// Remove existing variables
drop _all

//We first create the matrix to store the results of each study
set matsize 10000
matrix A = J(`totalobs',1,.)
matrix B = J(`totalobs',1,.)
matrix C = J(`totalobs',1,.)
matrix D = J(`totalobs',1,.)

        forvalues i = 1/`studies' {
        scalar lambdai = 0.5+30*runiform()
        scalar alphai = `alpha'+2*rnormal()

                forvalues j = 1/`estperstudy' {
                // STEP ONE: Create the data for each study and estimate an effect
                clear
                set obs 100
                generate x = rnormal()
                // Note that each "study" has a difference error variance, causing the estimate
                // of the effect to be estimated with varying degrees of precision
                // This is for random effects
                scalar lambdaij = lambdai+`theta'*runiform()
                scalar alphaij = alphai+0.5*rnormal()
                generate e = lambdaij*rnormal()
                generate y = 1 + alphaij*x + e
                // This is for fixed effects
                // scalar lambda = 0.2+30*runiform()
                // generate e = lambda*rnormal()
                // generate y = 1 + `alpha'*x + e
                quietly regress y x
                scalar coef = _b[x]
                scalar secoef = _se[x]
                scalar tcoef = coef/secoef
                scalar ID = `i'
                scalar obsno = (`i'-1)*`estperstudy'+`j'

                // First run this program once to get the pre-publication study sample data
                // To get post-publication study sample data, uncomment one of the two sections
                // below.
                matrix A[obsno,1] = coef
```

78

```
              matrix B[obsno,1] = secoef
              matrix C[obsno,1] = tcoef
              matrix D[obsno,1] = ID
              }

       }
```

// The next set of commands moves the data out of matrices and reformats them as
// standard Stata data series. We have now completed generating our individual
// studies and we now move into the "meta-analysis" stage.
```
matrix bob = A,B,C,D
svmat bob
rename bob1 effect
rename bob2 seeffect
rename bob3 teffect
rename bob4 ID
generate pet = (1/seeffect)
```

// This set of commands imposese the publication bias, where the publication criterion
// is either that (i) the t-stat must be greater than or equal to 2, or (ii) the
// estimated effect is positive. The commands below implement the assumption that a
// study must have at least 7 out of 10 estimates that satisfy the publication
// criterion in order for the study to be "published."
```
generate dummy = 1
```
<mark>//replace dummy = 0 if abs(teffect) < 2</mark>
<mark>//replace dummy = 0 if effect < 0</mark>
```
by ID, sort: egen select = mean(dummy)
```
// Not sure why this happens, but if I put select<0.70, it kicks out the studies that
// have 7 estimates that satisfy the publication criterion. So I set select<0.65.
// Studies are omitted from the "meta-analysis" sample by replacing the relevant variables
// with missing values.
```
replace effect = cond(select<0.65,.,effect)
replace seeffect = cond(select<0.65,.,seeffect)
replace teffect = cond(select<0.65,.,teffect)
replace pet = cond(select<0.65,.,pet)
```

// This creates dummy variables for each of the 30 studies
// The dummy variables take names dum1 to dum30
```
tab ID, gen(dum)
```

```
       // This creates study-specific SE terms for use in the PEESE
       // according to equation 5.7 on page 85 of S&D
       forvalues i = 1/30 {
       generate SE`i' = seeffect*dum`i'
       }
```

// This estimate produces the OLS estimate of the effect
```
regress effect
return scalar effect_OLS = _b[_cons]
test _b[_cons] = `alpha'
return scalar pvalue_OLS = r(p)
```

// This estimate produces the PET version of the "publication bias"-
// corrected effect estimate
// NOTE1: According to equation 5.6 on page 85 of S&D, the bias-corrected
// effect is given by the coefficients on the respective precision terms, pet*.
// The specification below forces all the effects to be the same, while

// allowing for fixed effects to correct for bias-associated with estimate SEs.
// NOTE2: Also note that while all the dummy variables will not be included in the
// meta-analysis sample, this is not a problem because STATA will automatically
// kick out the ones that don't belong.
```
regress teffect dum1-dum30 pet, vce(cluster ID)
return scalar effect_PET = _b[pet]
test pet = `alpha'
return scalar pvalue_PET = r(p)
```

// This estimate produces the PEESE version of the "publication bias"-
// corrected effect estimate. It is based on equation 5.7 on page 85 of S&D.
// See notes from above.
```
regress teffect SE1-SE30 pet, noc vce(cluster ID)
return scalar effect_PEESE = _b[pet]
test pet = `alpha'
return scalar pvalue_PEESE = r(p)
```

// This estimate produces the FE estimate of the effect
```
generate constant = 1
vwls effect constant, sd(seeffect) nocon
return scalar effect_FE = _b[constant]
test _b[constant] = `alpha'
return scalar pvalue_FE = r(p)
```

// This estimate produces the WLS estimate of the effect
```
regress teffect pet, noc
return scalar effect_WLS = _b[pet]
test pet = `alpha'
return scalar pvalue_WLS = r(p)
```

// This estimate produces the RE estimate of the effect
// NOTE: We use the Method of Moments (mm) option of metareg
// because the maximum likelihood procedure had too many instances
// of failure to optimize. Method of Moments does not require
// iteration and thus avoids this problem.
```
metareg effect , wsse(seeffect) mm
matrix bill = e(b)
return scalar effect_RE = bill[1,1]
test _cons = `alpha'
return scalar pvalue_RE = r(p)
```

// This last command keeps track of how many estimates are in our "meta-analysis"
```
return scalar N = e(N)
```

```
end
```

.do file for TABLES 8 and 9 – Part B

```
// This program takes about 5 hours to run on my computer
etime, start
drop _all
clear
graph drop _all
set more off
set seed 52256
matrix EFFECT = J(9,7,0)
matrix MSE = J(9,7,0)
matrix TEST = J(9,7,0)
local studies = 30
local estperstudy = 10
local totalobs = `studies'*`estperstudy'
local theta = 1
local obs = 100

local i = 1

        foreach alpha in 0 0.5 1 1.5 2 2.5 3 3.5 4 {
        simulate effect_OLS = r(effect_OLS) effect_FE = r(effect_FE) effect_RE = r(effect_RE) ///
        effect_PET = r(effect_PET) effect_PEESE = r(effect_PEESE)  ///
        pvalue_OLS = r(pvalue_OLS) pvalue_FE = r(pvalue_FE) pvalue_RE = r(pvalue_RE) ///
        pvalue_PET = r(pvalue_PET) pvalue_PEESE = r(pvalue_PEESE) N = r(N) ///
        effect_WLS = r(effect_WLS) pvalue_WLS = r(pvalue_WLS) , ///
        reps(1000): PANELpbias, studies(`studies') estperstudy(`estperstudy') ///
        totalobs(`totalobs') alpha(`alpha') theta(`theta') obs(`obs')

        summ N, meanonly
        matrix EFFECT[`i',1] = r(mean)
        summ effect_OLS, meanonly
        matrix EFFECT[`i',2] = r(mean)
        summ effect_PET, meanonly
        matrix EFFECT[`i',3] = r(mean)
        summ effect_PEESE, meanonly
        matrix EFFECT[`i',4] = r(mean)
        summ effect_FE, meanonly
        matrix EFFECT[`i',5] = r(mean)
        summ effect_WLS, meanonly
        matrix EFFECT[`i',6] = r(mean)
        summ effect_RE, meanonly
        matrix EFFECT[`i',7] = r(mean)

        summ N, meanonly
        matrix MSE[`i',1] = r(mean)
        generate mse_OLS = (effect_OLS - `alpha')^2
        summ mse_OLS, meanonly
        matrix MSE[`i',2] = r(mean)
        generate mse_PET = (effect_PET - `alpha')^2
        summ mse_PET, meanonly
        matrix MSE[`i',3] = r(mean)
        generate mse_PEESE = (effect_PEESE - `alpha')^2
        summ mse_PEESE, meanonly
        matrix MSE[`i',4] = r(mean)
        generate mse_FE = (effect_FE - `alpha')^2
```

81

```
            summ mse_FE, meanonly
            matrix MSE[`i',5] = r(mean)
            generate mse_WLS = (effect_WLS - `alpha')^2
            summ mse_WLS, meanonly
            matrix MSE[`i',6] = r(mean)
            generate mse_RE = (effect_RE - `alpha')^2
            summ mse_RE, meanonly
            matrix MSE[`i',7] = r(mean)


             summ N, meanonly
             matrix TEST[`i',1] = r(mean)
            generate RROLS = 0
            replace RROLS = cond(pvalue_OLS<0.05,1,0)
            summ RROLS, meanonly
            matrix TEST[`i',2] = r(mean)
            generate RRPET = 0
            replace RRPET = cond(pvalue_PET<0.05,1,0)
            summ RRPET, meanonly
            matrix TEST[`i',3] = r(mean)
            generate RRPEESE = 0
            replace RRPEESE = cond(pvalue_PEESE<0.05,1,0)
            summ RRPEESE, meanonly
            matrix TEST[`i',4] = r(mean)
            generate RRFE = 0
            replace RRFE = cond(pvalue_FE<0.05,1,0)
            summ RRFE, meanonly
            matrix TEST[`i',5] = r(mean)
            generate RRWLS = 0
            replace RRWLS = cond(pvalue_WLS<0.05,1,0)
            summ RRWLS, meanonly
            matrix TEST[`i',6] = r(mean)
            generate RRRE = 0
            replace RRRE = cond(pvalue_RE<0.05,1,0)
            summ RRRE, meanonly
            matrix TEST[`i',7] = r(mean)

            local `++i'
            }

matrix colnames EFFECT = N OLS PET PEESE FE WLS RE
matrix rownames EFFECT = A0 A0P5 A1 A1P5 A2 A2P5 A3 A3P5 A4
matrix colnames MSE = N OLS PET PEESE FE WLS RE
matrix rownames MSE = A0 A0P5 A1 A1P5 A2 A2P5 A3 A3P5 A4
matrix colnames TEST = N OLS PET PEESE FE WLS RE
matrix rownames TEST = A0 A0P5 A1 A1P5 A2 A2P5 A3 A3P5 A4
matrix list EFFECT
matrix list MSE
matrix list TEST

etime
```

.do file for FIGURE 7 – Part A

<u>NOTE</u>: The yellow-highlighted part of the code determines which type of publication bias is operative

```
set more off
program drop _all
program define PANELfig, rclass
version 13.1

syntax, studies(integer) estperstudy(integer) totalobs(integer) alpha(real) ///
theta(real) obs(integer)

// Remove existing variables
drop _all

//We first create the matrix to store the results of each study
set matsize 10000
matrix A = J(`totalobs',1,.)
matrix B = J(`totalobs',1,.)
matrix C = J(`totalobs',1,.)
matrix D = J(`totalobs',1,.)

        forvalues i = 1/`studies' {
        scalar lambdai = 0.5+30*runiform()
        scalar alphai = `alpha'+2*rnormal()

                forvalues j = 1/`estperstudy' {
                // STEP ONE: Create the data for each study and estimate an effect
                clear
                set obs 100
                generate x = rnormal()
                // Note that each "study" has a difference error variance, causing the estimate
                // of the effect to be estimated with varying degrees of precision
                // This is for random effects
                scalar lambda = lambdai+`theta'*runiform()
                scalar alphaij = alphai+0.5*rnormal()
                generate e = lambda*rnormal()
                generate y = 1 + alphaij*x + e

                quietly regress y x
                scalar coef = _b[x]
                scalar secoef = _se[x]
                scalar tcoef = coef/secoef
                scalar ID = `i'
                scalar obsno = (`i'-1)*`estperstudy'+`j'

                // First run this program once to get the pre-publication study sample data
                // To get post-publication study sample data, uncomment one of the two sections
                // below.
                matrix A[obsno,1] = coef
                matrix B[obsno,1] = secoef
                matrix C[obsno,1] = tcoef
                matrix D[obsno,1] = ID
```

```
            }
        }
```

// The next set of commands moves the data out of matrices and reformats them as
// standard Stata data series. We have now completed generating our individual
// studies and we now move into the "meta-analysis" stage.
matrix bob = A,B,C,D
svmat bob
rename bob1 effect
rename bob2 seeffect
rename bob3 teffect
rename bob4 ID
generate pet = (1/seeffect)

// This set of commands imposese the publication bias, where the publication criterion
// is either that (i) the t-stat must be greater than or equal to 2, or (ii) the
// estimated effect is positive. The commands below implement the assumption that a
// study must have at least 7 out of 10 estimates that satisfy the publication
// criterion in order for the study to be "published."
generate dummy = 1
//replace dummy = 0 if abs(teffect) < 2
//replace dummy = 0 if effect < 0
by ID, sort: egen select = mean(dummy)
// Not sure why this happens, but if I put select<0.70, it kicks out the studies that
// have 7 estimates that satisfy the publication criterion. So I set select<0.70.
// Studies are omitted from the "meta-analysis" sample by replacing the relevant variables
// with missing values.
replace effect = cond(select<0.65,.,effect)
replace seeffect = cond(select<0.65,.,seeffect)
replace teffect = cond(select<0.65,.,teffect)
replace pet = cond(select<0.65,.,pet)

// This creates dummy variables for each of the 30 studies
// The dummy variables take names dum1 to dum30
tab ID, gen(dum)

        // This creates study-specific SE terms for use in the PEESE
        // according to equation 5.7 on page 85 of S&D
        forvalues i = 1/30 {
        generate SE`i' = seeffect*dum`i'
        }

// This estimate produces the OLS estimate of the effect
regress effect
return scalar effect_OLS = _b[_cons]

// This estimate produces the FE estimate of the effect
generate constant = 1
vwls effect constant, sd(seeffect) nocon
return scalar effect_FE = _b[constant]

// This estimate produces the RE estimate of the effect
// NOTE: We use the Method of Moments (mm) option of metareg
// because the maximum likelihood procedure had too many instances
// of failure to optimize. Method of Moments does not require
// iteration and thus avoids this problem.
metareg effect , wsse(seeffect) mm

```
matrix bill = e(b)
return scalar effect_RE = bill[1,1]

// This estimate produces the PET version of the "publication bias"-
// corrected effect estimate
// NOTE1:  According to equation 5.6 on page 85 of S&D, the bias-corrected
// effect is given by the coefficients on the respective precision terms, pet*.
// The specification below forces all the effects to be the same, while
// allowing for fixed effects to correct for bias-associated with estimate SEs.
// NOTE2: Also note that while all the dummy variables will not be included in the
// meta-analysis sample, this is not a problem because STATA will automatically
// kick out the ones that don't belong.
regress teffect dum1-dum30 pet, vce(cluster ID)
return scalar effect_PET = _b[pet]

// This estimate produces the PEESE version of the "publication bias"-
// corrected effect estimate.  It is based on equation 5.7 on page 85 of S&D.
// See notes from above.
regress teffect SE1-SE30 pet, noc
return scalar effect_PEESE = _b[pet]

end
```

.do file for FIGURE 7 – Part B

NOTE: **The yellow-highlighted part of the code determines which type of publication bias is operative**

```
// This program takes approximately 3 minutes to run
etime, start
drop _all
clear
graph drop _all
set more off
set seed 52256
matrix EFFECT = J(9,6,0)
matrix MSE = J(9,6,0)
matrix TEST = J(9,6,0)
local studies = 30
local estperstudy = 10
local totalobs = `studies'*`estperstudy'
local theta = 1
local obs = 100

local i = 1

        foreach alpha in 1  {
        simulate effect_FE = r(effect_FE) effect_RE = r(effect_RE) effect_OLS = r(effect_OLS) ///
        effect_PET = r(effect_PET) effect_PEESE = r(effect_PEESE), ///
        reps(100): PANELfig, studies(`studies') estperstudy(`estperstudy') totalobs(`totalobs') ///
        alpha(`alpha') theta(`theta') obs(`obs')

        local `++i'
        }

kdensity effect_OLS , xline(1) name(AllMeta) addplot(kdensity effect_PET ||  kdensity effect_PEESE ///
|| kdensity effect_FE || kdensity effect_RE)

etime
```