UNIVERSITY OF CALIFORNIA, MERCED

A Weight-Function Model for Moderators of Publication Bias

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Psychological Sciences

by

Kathleen M. Coburn

2018

Committee in charge:
Professor Jack L. Vevea, Chair
Professor Sarah Depaoli
Professor Rose Scott

The Dissertation of Kathleen M. Coburn is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Jack Vevea Chair

Sarah Depaoli

Rose Scott

University of California, Merced
2018

To Kyle Hamilton.

Table of Contents

List of Abbreviations and Symbols

Although all abbreviations and symbols are defined when they first appear in the text, I outline a list of the most frequently used here for convenience. Abbreviations and symbols are listed in order of their first appearance within the text.

1. I^2: an index of the proportion of total heterogeneity that is between-studies
2. EBE: "Exotic Becomes Erotic" theory
3. *JPSP*: *Journal of Personality and Social Psychology*
4. GUI: graphical user interface
5. JAGS: Just Another Gibbs Sampler
6. *d*: effect sizes representing standardized mean differences
7. *k*: the number of studies in a meta-analytic dataset
8. *BMJ*: the *British Medical Journal*
9. τ^2: variance component, measure of heterogeneity between studies
10. σ^2: vector of sampling variances
11. *r*: rate parameter of an exponential distribution
12. *prob*: probability
13. 2PL: two-parameter logistic model
14. IRT: item response theory
15. Θ: theta, a latent trait in item response theory
16. *a*: parameter in item response theory representing the maximum slope of the curve (the discrimination parameter)
17. *b*: parameter in item response theory representing the location on the *x*-axis where the curve is centered (the difficulty parameter)
18. 3PL: three-parameter logistic model
19. λ : parameter in the lambda model representing information about the difference in the selection-bias pattern across levels of a dichotomous variable
20. Y_i: a vector of effect sizes
21. $\phi(x)$: standard normal probability density function evaluated at *x*
22. *q*: number of moderator variables in a meta-analytic model
23. $\boldsymbol{\beta}$: a *q*-dimensional vector of unknown regression coefficients
24. $\boldsymbol{X}$: a matrix of known moderator variables
25. $\boldsymbol{X}_i$: a vector of a known moderator variable
26. Δ_i: $\boldsymbol{X_i\beta}$, or a function of linear predictors
27. *w(x)*: a weight function
28. $\boldsymbol{Z}$: a matrix of scores representing group membership across levels of a relevant moderator of publication bias
29. $\boldsymbol{Z}_i$: a vector of scores representing group membership in one level of a relevant moderator of publication bias
30. ω : a vector of weight parameters corresponding to given *p*-value intervals
31. *Z*-statistic: effect sizes divided by their standard errors
32. CI: confidence interval
33. $\boldsymbol{\beta_0}$: the model intercept, or the mean of the meta-analytic dataset if there are no predictors
34. *U(x, y):* uniform distribution with a lower bound of x and an upper bound of y

35. $\phi(x)$: a Poisson observation of x
36. log x: natural logarithm of x
37. $N(x, y)$: normal distribution with a mean of x and a standard deviation of y
38. $\Gamma(x, y)$: gamma distribution with a shape parameter of x and a rate parameter of y
39. MCMC: Markov chain Monte Carlo
40. PSRF: potential scale reduction factor(s)
41. *M*: minimum number of burn-in iterations required by the Raftery and Lewis diagnostic
42. *N*: minimum number of post-burn-in iterations required by the Raftery and Lewis diagnostic
43. *mn*: notation for the mean, or intercept, in Bayesian estimation
44. *vcinv*: notation for the variance component in Bayesian estimation
45. CRAN: Comprehensive *R* Archive Network
46. GATB: the General Aptitude Test Battery

List of Figures

List of Tables

Acknowledgements

This dissertation is dedicated to Kyle Hamilton, who comforts me, entertains me, supports me, and above all else loves me. There are not enough words in the world to thank you.

Next, I would like to thank Dr. Jack L. Vevea for taking me into his lab, for supporting me and motivating me throughout my graduate career, and for being not only an excellent advisor and a fantastic role model but also a friend. I am so glad I met you, Jack, and that you accepted me as a student. I truly enjoyed working with you and learning from you, and without you I would not be here today. I would also like to thank the other two members of my committee, Dr. Sarah Depaoli and Dr. Rose Scott, whose support and feedback throughout my graduate school career have been invaluable.

Thanks go as well to my brother Andrew Arnold, who is one of the best men I've ever known. I also want to thank my sister-in-law Tara Walsh, who is equally inspirational, and my niece and nephews, Grace, Louis, and Sylas, for their smiles, laughter, and love. I want to thank Sharon and Eric Hamilton for welcoming me and for treating me like a member of the family, which means more to me than words can express. (I also thank Lauren Hamilton for her support and wish her the best of luck on her own educational journey!)

Finally, and last but not least, I want to thank my dogs Maggie and Mona for their consistent and unconditional love, affection, and companionship. Mona crossed the Rainbow Bridge during the completion of this dissertation. She passed away not as a possession but as a family member, not forgotten but loved with all our hearts.

Vita

Education

Ph. D in Psychological Sciences

University of California, Merced, Merced, CA, USA 2012 – 2018 (Expected)

Bachelor of Science in Psychology

San Jose State University, San Jose, CA, USA 2007 – 2011

Publications

Vevea, J. L. & **Coburn, K. M.** (in press). Publication bias. In Cooper, H., Hedges, L. V., & Valentine, J. C. (Eds.), *Handbook of Research Synthesis and Meta-Analysis*. Russell Sage Foundation.

Coburn, K. M. & Vevea, J. L. (2015). Publication bias as a function of study characteristics. *Psychological Methods*, *20*(3), 310-330.

Vevea, J. L. & **Coburn, K. M.** (2015). Maximum-likelihood methods for meta-analysis: A tutorial using *R*. *Group Processes and Intergroup Relations*, *18*(3), 329-347.

Abstract of the Dissertation

A Weight-Function Model for Moderators of Publication Bias

by

Kathleen M. Coburn

Doctor of Philosophy

University of California, Merced, 2018

Professor Jack L. Vevea, Chair

This dissertation begins by demonstrating that publication bias can depend on factors other than statistical significance, including study characteristics like social preferences and source of funding. After providing an empirical example of differing bias patterns, the dissertation presents a weight-function model that is capable of accommodating moderators of publication bias. Subsequent chapters describe a version of the model designed for sensitivity analyses, a Bayesian version, and an *R* package for implementing the model. Throughout the dissertation, the performance of each version of the model is assessed via simulation. Overall, the model outperforms competing models across simulation cells, and appears to be an effective tool in cases where study characteristics affect publication bias.

Chapter 1: An Introduction to Methods of Assessing Publication Bias

Across many scientific disciplines, regardless of subject, results favoring the researchers' aims are vastly more likely to be published (Begg & Berlin, 1989; Fanelli, 2012; Ioannidis, 1998). These favorable results are often termed "positive," while unfavorable results, which do not support researchers' hypotheses and are usually not statistically significant, are "negative" (Fanelli, 2012). In this dissertation, I disregard any directional claim and distinguish results as *favorable* or *unfavorable*. I do so for two reasons: First, calling all favorable results positive is an oversimplification, ignoring the fact that many researchers are interested in *negative* effects. An effect size may be "positive" to one researcher while simultaneously "negative" to another. Second, it seems counterintuitive to refer to results we aim to encourage in the research literature with such an emotionally laden word as "negative."

Publication bias typically arises when favorable results are published, while unfavorable results remain languishing in investigators' file drawers (Iyengar & Greenhouse, 1988; Robert Rosenthal, 1979; Scargle, 1999). This situation is often nicknamed "the file-drawer problem" (Rosenthal, 1979). Of course, in reality, most investigators likely do not have file cabinets bursting with unfavorable, neglected papers; if a project begins to appear unfavorable, it may not receive funding or even be completed, much less written up. Fundamentally, publication bias can arise from any situation in which the body of published literature on an effect is not reflective of the true effect. Research with unfavorable conclusions, for any given reason, remains unpublished and, therefore, unavailable.

The popularity of meta-analyses and quantitative systematic reviews has helped increase researchers' awareness of publication bias (Parekh-Bhurke et al., 2011), partially because researchers can observe the effects of bias more easily when they attempt to combine effect sizes on a subject. Meta-analyses are often described as a more valid means of reviewing research than a narrative or qualitative review (King & He, 2006; Rothstein, 2008), but if meta-analysts cannot retrieve a sample of studies that is not systematically biased, the validity of their results may be called into question, along with any substantive conclusions they draw (Rothstein, 2008). It is important to realize, however, that publication bias is *not* a problem *of* meta-analysis, but rather a problem of the research community as a whole. Meta-analysis has increased our awareness of the issue and offers researchers the best opportunity of assessing the magnitude and severity of bias.

1.1 Meta-Analysis

Because assessing publication bias is possible in the context of meta-analysis, it is logical to discuss meta-analysis before discussing methods of bias assessment. This section provides a general overview of meta-analysis methodology, not guidelines for conducting a meta-analysis, so details are limited. Many texts (including, but not limited to, Lipsey & Wilson, 2001; Cooper, Hedges, & Valentine, 2009; Borenstein, Hedges,

Higgins, & Rothstein, 2011; Schmidt & Hunter, 2014; Cumming, 2013) provide thorough further instruction for interested readers.

In the twenty-first century, as consumers of research face an explosion of publicized scientific findings, meta-analysis may become a more useful tool than ever, and rightfully so. Users log on to social media daily only to be faced with a barrage of claims, and it may be difficult or even impossible for them to identify which are plausible and which are not. The same newspaper or Web site which one day proclaims that sugar causes excessive weight gain may encourage sugar consumption as a diet supplement the next day. If each of these conclusions is based on the results of a different study, which one should users heed? An individual study and its corresponding statistical significance is rarely the best source of information about population effects. Although a small literature does argue that single, well-designed, large studies should be the gold standard under some contexts (Scifres, Iams, Klebanoff, & Macones, 2009; LeLorier, Gregoire, Benhaddad, Lapierre, & Derderian, 1997), meta-analyses are superior in most contexts. Meta-analysis combines multiple sources to yield an overall effect estimate, resulting in higher power than a single study, and even allows the use of study characteristics as moderators, so that the effect may vary across study types (Rosenthal & DiMatteo, 2001).

There is evidence of attempts at research synthesis dating back to the early 19th century, mostly involving the reconciliation of differing estimates of physical constants (Chalmers, Hedges, & Cooper, 2002; Stigler, 1986; Nichols, 1891). However, meta-analysis in a more familiar form emerged in 1904 with Karl Pearson, who synthesized correlation coefficients to assess the efficacy of smallpox inoculation, finding an average correlation between inoculation and survival of about 0.63 (Pearson, 1904; Rosenthal & DiMatteo, 2001). The medical fields conducted meta-analyses before the social sciences, likely because they were the first to be faced with an overwhelming body of evidence needing to be synthesized (Rosenthal & DiMatteo, 2001). Psychology joined in the late 1970s; perhaps the most famous of the early psychology meta-analyses was that conducted by Smith, Glass, and Miller (1980) on psychotherapy effectiveness. (Glass, in fact, initially coined the term "meta-analysis" in 1976.)

To conduct a meta-analysis, researchers must first identify the question they wish to answer. Meta-analysis is sufficiently powerful and capable of addressing virtually any question that researchers can address in an individual study, even incorporating mediation, moderation, or structural equation modeling if necessary. The process of identifying a question should lead researchers to determine their study qualifications – that is, what features must a study have to be included, and what features will require that a study be omitted. Often, at this stage, the researcher may have decided what specific study characteristics are of interest and created a corresponding coding manual. Such characteristics might include the age of study participants, the location where the study was conducted, or any other potential moderators, ranging from the type of treatment to the type of outcome measure. After identifying a question and creating a coding manual, the researcher must begin the always-tedious process of searching the literature for results to incorporate and extracting and coding the necessary information from those studies.

Once the researcher has identified studies and calculated effect sizes from those studies, he or she may begin analyzing the data using a fixed-effect, random-effects, or mixed-effects meta-analytic model. The researcher should choose a model based on the nature of the intended universe of generalization – that is, whether or not inference will

extend beyond the observed studies (Hedges & Vevea, 1998). If the researcher wishes to make inferences about a population beyond the studies at hand, a random- or mixed-effects model is appropriate (Hedges & Vevea, 1998). In the rare case where a researcher's goal is to generalize to exactly the population of studies at hand, addressing only the uncertainty of sampling in those studies, then a fixed-effect model is appropriate (Hedges & Vevea, 1998).

The primary difference between fixed-effect, random-effects, and mixed-effects meta-analytic models has to do with their handling of sources of variation between and among studies. Each study yields an effect size and a corresponding sample size. Effect sizes within studies, even if those studies are exact replications, will vary; this variation is primarily determined by sample size, and is sometimes referred to as sampling variability, or within-studies heterogeneity. This variation is the reason why two independent studies, each with 10,000 participants, are likely to produce slightly different effect-size estimates, although both are measuring the same phenomenon. A fixed-effect meta-analytic model allows only for the presence of within-studies heterogeneity and assumes that all studies in the meta-analysis are assessing an identical underlying population effect[1].

Random-effects meta-analytic models allow for differences between the population effects of different studies. Two studies that represent different populations usually produce different effect sizes. A meta-analysis is not likely to contain representations of all possible populations. In such a case, using a fixed-effect model would allow the researcher to draw conclusions only about the two specific populations featured in their dataset. A random- or mixed-effects model, on the other hand, would allow the researcher to make inferences about studies involving all other possible populations. In short, random-effects meta-analyses allow for the possibility that studies are estimates of slightly different underlying population effects. This variation between studies (or between-studies heterogeneity) is measured by a parameter called the variance component, usually denoted τ^2.

This section is a brief summary of a complex procedure, but it has explained the necessary features to introduce the rest of this dissertation.

1.2 Publication Bias

There are many tools available for assessing publication bias. Some, like the selection models, can yield estimates of the mean effect(s) and the variance component that are adjusted for bias; some perform significance tests for the presence of bias; some do both of the above. For a detailed summary of these tools and examples of their use, I invite readers to peruse the related chapter in the *Handbook of Research Synthesis and Meta-analysis* (Vevea & Coburn, in press). Other authors have summarized these bias assessments as well; for books, I recommend *Publication Bias in Meta-Analysis: Prevention, Assessment and Adjustments* (Rothstein, Sutton, & Borenstein, 2005). Many articles also summarize these bias assessments (for example, see Peters et al., 2006; Ferguson & Brannick, 2012; Macaskill, Walter, & Irwig, 2001; Sterne, Egger, & Smith,

[1] An exception is the "fixed-effects" model. In this case, the concern is only with within-study variability, but the goal is to estimate several average population effects (Rice, Higgins, & Lumley, 2017). The circumstances in which such a model might be useful are unclear.

2001). Therefore, the descriptions provided in this chapter are relatively brief, and interested readers are advised to investigate these citations.

Readers may notice that a common technique, known as the failsafe-*N*, is absent from this discussion. The failsafe-*N* (Rosenthal, 1979) is excluded for a number of reasons. Although the technique is both intuitively appealing and popular among meta-analysts, it is utterly valueless (Vevea & Coburn, in press); it has neither an underlying statistical model nor any kind of criterion, it is largely arbitrary, and it cannot accommodate empirical situations like between-studies heterogeneity (Vevea & Coburn, in press; Begg & Berlin, 1988; Becker, 2005; Iyenger & Greenhouse, 1988).[2] Presenting a substantive example of the failsafe-*N*, even with a strong caveat, would promote use of the technique, and as an ethical researcher I cannot promote the use of the failsafe-*N*.

1.3 Visual Methods

This section discusses two visual methods of assessing data for the presence of publication bias. It is important to note that neither of these methods can produce adjusted estimates of parameter values – that is, neither can quantitatively measure the impact of publication bias. These methods also cannot produce a statistical test for the presence of publication bias. By nature, interpretation of both graphs is somewhat subjective. However, they provide a crucial first glance at data patterns, and may inform future bias-related analyses.

1.3.1 Funnel Plots

Light and Pillemer (1984) originally introduced the funnel plot as a scatterplot with effect-size estimates on the horizontal axis and their corresponding sample sizes on the vertical axis. The expectation, in the absence of both systematic heterogeneity and publication bias, is that the effect sizes will be evenly distributed around their underlying population mean (Light & Pillemer, 1984; Vevea & Coburn, in press). There will be more variability among the effect sizes with smaller sample sizes, due to the greater influence of sampling error. This means that, ideally, the scatterplot (if no publication bias is present) will appear symmetric with respect to the distribution of effect sizes and will resemble a funnel (Vevea & Coburn, in press).

In the years since its introduction, variations of the funnel plot have gained popularity. Researchers have debated whether to use sample size or another measure of study precision, such as variance or standard error (Vevea & Hedges, 1995; Sterne & Egger, 2001; Vevea & Coburn, in press). Sterne and Egger (2001) presented comparative plots using different measures of precision, noting that the choice of measure can impact interpretation of the plot. In this dissertation, I plot effect sizes against standard errors; this method allows the distribution of effect sizes to cover more plot space for the smaller studies, among which publication bias is more likely to be evident (Vevea & Coburn, in press). I also plot effect-size estimates on the vertical axis and standard errors on the

[2] Orwin (1983) produced a modified version of the failsafe-*N* which does not assume that all missing effect sizes average to zero, and this version should not be condemned quite as strictly.

horizontal axis, to remain consistent with the graphical convention of plotting unknown quantities on the Y-axis and fixed quantities on the X-axis (Vevea & Coburn, in press).

When interpreting funnel plots, it is important to remember that asymmetry in the plot may be due to phenomena other than publication bias; *any* influence that is associated with both study precision and effect size can result in asymmetry (Vevea & Coburn, in press). Moderators of effect size, or systematic heterogeneity, can create asymmetry that is not due to bias. Because other factors can cause asymmetry, and because funnel plot interpretation is subjective by nature, the utility of the funnel plot is sometimes questioned (Terrin et al., 2005; Lau et al. 2006; Tang & Liu, 2000; Hunter et al., 2014). Nevertheless, the plot has value, particularly as a preliminary assessment of the presence of bias.

1.3.2 Cumulative Meta-Analysis

The technique of cumulative meta-analysis is often used to determine whether a meta-analytic mean appears to stabilize in relation to some variable of interest; originally, the variable was time of publication (Clarke, Brice, & Chalmers, 2014). To visually assess whether the mean stabilizes, the meta-analyst creates a plot of the results. In such a plot, horizontal lines are drawn representing the pooled mean effect sizes and their corresponding error bars; the lowest line represents the pooled estimate of one effect size, the next two, and so on. The bottom line of the plot then represents the overall meta-analysis including all of the effect-size estimates.

Kepes et al. (2014) demonstrated that it is possible to use cumulative meta-analysis to explore the presence of publication bias by sorting effect sizes by a measure of study precision, rather than by year of publication. Studies can be pooled in order from least precise to most precise, or from most to least. After sorting, if the average effect size drifts (becoming either more positive or negative) as studies are added, there is evidence of a relationship between precision and effect size; in other words, publication bias may be present. Of course, like the funnel plot and the following methods based on the funnel plot, relationships between study size and effect size may exist due to moderator variables rather than publication bias. Moreover, for some effect measures (e.g., log odds ratios), there is some inherent association between effect size and sample size (Ialongo, 2016). Therefore, if such a relationship does exist, it should be interpreted with caution.

1.4 Statistical Tests Based on Visual Methods

This section presents a brief discussion of three methods for assessing publication bias that yield statistical tests for the presence of a relationship between some measure of study size and effect size. All three methods are based on some variation of the funnel plot.

It is important to remember that, although the relationship between study size and effect size can be viewed as a proxy for the presence of publication bias, such a relationship may exist when bias is not present; it may be due to moderator variables. These tests assume homogeneity of effect sizes; they assume that any asymmetry is due to bias. If the collection of effects is heterogeneous (i.e., has a large variance component),

these tests may show that bias is present when it is not. As with all statistical tests, they should be used (and interpreted) with care.

1.4.1 Rank Correlation

The rank correlation was not originally developed to address publication bias; however, Begg and Mazumdar (1994) discovered that it provided a formal test for asymmetry in funnel plots. Their test calculates a rank correlation between the deviations of the effect sizes from their mean and the sampling variances of the effect sizes (primarily determined by study size). The correlation is a test statistic that can then be compared to the standard normal distribution; significance indicates the presence of a relationship between study size and effect size may be present.

1.4.2 Egger's Linear Regression

This test may be intuitive; if a test of the correlation between study size and effect size is useful, a regression of effect size on study size (or precision) is likely also useful. Egger et al. (1997) first described this process; they regressed effect sizes' standard normal deviates (effect sizes divided by standard errors) on their precision (defined as the inverse of the standard error). If the resulting regression line does not run through the origin – that is, if the intercept of the regression differs from zero – there is evidence of a relationship between study size and effect size (Egger et al., 1997)

There are several existing variations of the regression test for publication bias, which are not described here for the sake of brevity. Some switch the role of intercept and slope, some involve different measures of study size, some accommodate binary outcomes, and so on. I refer interested readers to Vevea and Coburn (in press), which can, at the very least, provide a reading list.

1.4.3 Trim-and-Fill

Trim-and-fill is a nonparametric method that was developed as a simpler alternative to parametric selection models (Duval and Tweedie, 2000a; 2000b). It is based on the idea that asymmetry present in funnel plots can be measured and rectified by imputing the "missing" effect sizes. The method uses an iterative process to "trim" asymmetric portions of the plot, then uses one of three estimators to generate, or "fill in," new effect sizes that are mirror images of those that remain (Duval and Tweedie, 2000a; 2000b), reflected across the axis of the funnel plot. This results in an artificially symmetric plot. Adjusted estimates of the mean and variance component can be calculated based on this symmetric data set.

The trim-and-fill method, of course, has some flaws; it sometimes imputes very unrealistic effect sizes and can perform poorly in the presence of between-studies heterogeneity (see Vevea & Coburn, in prep, for a summary). Still, the method is popular and accessible.

1.5 Selection Models

This section presents an overview of two of the more well-known selection models for assessing publication bias. These models attempt to describe the mechanism for effect suppression and combine this with an effect-size model that describes the distribution of effects in the absence of publication bias. The key to these models is their assumptions about the mechanism of suppression. If there *was* a consistent (and known) method of suppression leading to the presence of publication bias, it would be relatively easy to model. Instead, researchers must make assumptions and theorize about the most likely process of selection. The choice of selection mechanism is typically where these selection models differ.

Although selection models are more complex and difficult to implement, they are usually recommended over other methods if the number of studies in the meta-analysis is adequate to support estimation. They are often capable of accommodating between-study heterogeneity and moderators of effect size, a feature that most other methods lack.

1.5.1 Vevea and Hedges

The Vevea and Hedges (1995) model is an extension of the original Hedges (1992) model. It assumes that the mechanism of selection is a step function based on *p*-values. The meta-analyst can specify a series of *p*-value cutpoints at perceived milestones of statistical significance – for instance, $p = 0.05$, $p = 0.10$, et cetera. The relationship is described as a step function because of the differences between one's perception of, for example, $p = 0.049$ and $p = 0.051$. Weights representing the relative likelihood of survival for effect sizes whose *p*-values fall in each of the *p*-value intervals can be estimated in the context of a fixed-effect, mixed-effects, or random-effects model. There is one exception: The first weight, which applies to the "most significant" range of *p*-values, is fixed at 1.0 to allow for model identification. This means that subsequent weights must be interpreted relative to the first weight; they may exceed 1.0 and are not directly interpretable as probabilities.

The Vevea and Hedges (1995) model provides estimates of the variance component, the mean effect (or the intercept and slopes of a linear model for the mean), and the vector of weights that represent the selection process. The simultaneous estimation of these weights has the effect of adjusting the other parameters for the presence of selection. A likelihood-ratio test is conducted, comparing the adjusted model including selection to its unadjusted fixed-effect, random-effects, or mixed-effects counterpart. If this test is significant, there is evidence that publication bias may be present.

1.5.2 Copas and Shi

The Copas and Shi model (2000; 2001) is frequently mentioned in discussions of selection models but has only recently seen practical use. It is a combination of a random-effects model of effect size and a selection model, in which the probability of effect-size survival is a linear function of its standard error. This linear function can be rewritten as a propensity model, in which an effect size can survive if and only if its propensity for

survival is greater than zero (Copas & Shi, 2001; Copas & Li, 1997). If there is a positive correlation between observed effect sizes and their estimated propensities, this indicates the presence of publication bias. A correlation of zero indicates that effect sizes survive regardless of their standard error – that is, the absence of bias. The need to fix some parameters for model identification means that this model is fundamentally a sensitivity analysis, rather than a full-blown estimation of adjusted effects.

For more details and a demonstration of the model, please see Vevea and Coburn (in press) or Carpenter et al. (2009).

1.6 Conclusions

This chapter has provided very general descriptions of the concept of meta-analysis and of several methods for assessing publication bias. Two of these methods, however, are especially relevant in the current dissertation. I recommend that readers pay specific attention to the funnel plot and the Vevea and Hedges (1995) weight-function model. The funnel plot appears in this dissertation several times, usually to aid readers in visualizing the presence and severity of asymmetry. The Vevea and Hedges (1995) model is the basis of the lambda model and its variations, which are the primary subject of this dissertation.

Chapter 2: Study Characteristics and Publication Bias

The methods in Chapter 1 assess situations in which studies with favorable, or statistically significant, results are more likely to be published. The problem of publication bias, however, is more complex than that. Publication bias does not depend solely on statistical significance. This chapter discusses the presence of other factors that can impact a study's likelihood of publication.

Imagine a situation in which two groups are conducting trials on the efficacy of a blood pressure medication. The first group has a conflict of interest, because the creators of the medication fund it. The second group also has a conflict: It is funded by the company producing the leading competing medication. If a given study's results are nonsignificant, indicating that the new medication is no better than the leading competition, the second group of researchers are probably much more interested in publishing the study than the first group. A meta-analyst who ignores the role of funding source may miss this pattern of bias and conclude that publication bias is not an issue, because the suppression of results in opposite directions cancel each other out.

Although funding source may not play as large a role in the social sciences as in the natural sciences or medical fields, researchers across fields conduct meta-analyses and should be aware of the role of funding in any case where it may be relevant. Studies are significantly more likely to favor a preferred therapy or treatment if they are funded by pharmaceutical firms or industry (Davidson, 1986; Easterbrook, Gopalan, Berlin, & Matthews, 1991; Kjaergard & Als-Nielsen, 2002; Lexchin, Bero, Djulbegovic, & Clark, 2016; Sismondo, 2008). In nutrition research focused on the relationship between beverages and health, industry-sponsored studies are significantly more likely to recommend their sponsors' products (Lesser, Ebbeling, Goozner, Wypij, & Ludwig, 2007). Industry funding has a similar effect on studies assessing the efficacy of nicotine replacement therapy for smoking cessation (Etter, Burri, & Stapleton, 2007), studies reviewing the health effects of mobile phone use (Huss, Egger, Hug, Huwiler-Müntener, & Röösli, 2007), and studies of the protective effects of alcohol on cardiovascular disease (McCambridge & Hartwell, 2015). A systematic review of passive smoking reviews found that the only factor associated with concluding that passive smoking is not harmful was whether an author was affiliated with the tobacco industry (Barnes & Bero, 1998). Industry-funded studies are also more likely to become publications if their results favor the industry's products (Melander et al., 2003). Bias related to source of funding can also operate in the other direction – for instance, "it should be borne in mind that there is currently considerable pressure from antismoking organizations for journals not to accept papers supporting any aspect of the tobacco industry's position on smoking and health, regardless of the scientific merits of the paper, presumably so that the message to smokers to give up will come over as clearly as possible" (Thornton & Lee, 2000, p. 209).

To further complicate the situation, funding source is not the only study characteristic that may influence publication, and there may even be interactions between study characteristics. Research indicates that factors including the gender of the first author, whether a study is single- or multi-center, whether or not a study is randomized, year of publication, language of publication, and social preferences can influence studies'

likelihood of publication, regardless of statistical significance (Abdel-Sattar, Krauth, Anglemyer, & Bero, 2014; Barnes & Bero, 1998; Begg & Berlin, 1988; Grégoire, Derderian, & Le Lorier, 1995; Kepes, Banks, McDaniel, & Whetzel, 2012; Melander, Ahlqvist-rastad, Meijer, & Beermann, 2003; Sismondo, 2008). This results in publication bias that depends on the level of the relevant study characteristic, since the magnitude of the average effect size differs across levels for no reason other than changing patterns of publication. For example, a randomized study must be prospectively organized, and therefore requires a greater time commitment and is more expensive than a comparative, non-randomized study (Begg & Berlin, 1988). This implies that randomized studies are more likely to be seen through to publication, regardless of whether or not their results are favorable (Begg & Berlin, 1988). Research confirms this theory, indicating that observational studies demonstrate a greater tendency toward publication bias than randomized studies (Berlin, Begg, & Louis, 1989; Dickersin & Min, 1993; Easterbrook et al., 1991). The trends for several types of therapy trials reveal that most nonrandomized studies demonstrated the efficacy of the new treatment, while most randomized trials reported no effect (Begg & Berlin, 1989). However, bias patterns related to randomization may differ as well, and randomization may interact with factors like funding source; some research shows that industry-funded randomized trials are more subject to bias than non-industry-funded ones (Djulbegovic et al., 2000).

Publication patterns can also vary across the type or prestige of journals. This effect is not necessarily due to bias on the part of journal editors or reviewers; it is possible, and understandably so, that researchers are likely to submit dramatic or exciting results in higher-prestige journals.

The time that research is conducted can also play a role in its likelihood of publication, for a variety of reasons. Earlier studies published on a topic tend to be described as "exploratory," while later ones on the same topic are "confirmatory." If an exploratory study is deemed uninteresting, or if its results are unfavorable, it is more likely to be discarded prior to publication (Begg & Berlin, 1989). Time can also interact with prevailing social preferences. One example is that of the decline in gender differences over time, which researchers speculate may be due to the publication of a popular book in 1974 that encouraged publication of nonsignificant results (Hyde & Linn, 1988). Sometimes, as in the case of gender differences, a time trend in effect sizes may be due to changing patterns of publication bias. This is also known as time-lag bias, or the idea that dissemination of studies through publication is faster for favorable results (Banks, Kepes, & Banks, 2012). In the early days of a field, that field may be more subject to publication bias as favorable studies are published quickly; later, more unfavorable or nonsignificant studies may be published. A difference in bias patterns over time can also be due to the fact that results, whether favorable or not, tend to be published sooner if they are dramatic (Banks et al., 2012). Time-lag bias, as described, also exists in meta-analyses of individual patient data and in clinical trials (Hopewell, Clarke, Stewart, & Tierney, 2007). The effects of time on publication vary across fields, types of study, and possibly even outcome measures. Meta-analysts should take care to consider their subject of interest and assess whether time may have influenced publication patterns.

For more information about the impact of study characteristics on publication bias patterns, and for an exploration of their effects on various empirical meta-analyses, I

invite the interested reader to peruse Coburn and Vevea (2015). The next section introduces a specific empirical dataset, impacted by a relevant study characteristic, that I will use throughout this dissertation to demonstrate the models presented.

2.1 Example

Throughout this dissertation, I use a substantive meta-analytic dataset to illustrate the issue of moderators of publication bias. This section introduces and describes the dataset; example sections in subsequent chapters use the dataset to demonstrate variations of the lambda model and the *R* package *weightr*. The dataset and a link to the corresponding meta-analysis are included in Appendix A.

2.1.1 Dataset

This dataset is from a meta-analysis conducted on precognition. It focuses on participants' ability to anticipate a randomly-occurring event before said event occurs. It consists of 90 studies; 9 are original experiments conducted by Daryl Bem, 69 are direct replications, and 11 studies are more generally related to the effects of randomly-occurring future events. Bem's original nine experiments (2011) assessed the retroactive effects of various stimuli, a concept known as "retrocausation." Several of these experiments were variations of a task in which participants were presented with two curtain images and asked to choose one; behind one curtain lay a blank wall, and behind the other lay some type of stimulus, such as erotic stimuli (representing a positive outcome). Another experiment assessed participants' ability to select a targeted image at random, during which they were either rewarded for a correct identification with a positively valenced image or penalized with a negatively valenced one. Most of the remaining experiments employed variations on this task to determine whether the roles of such psychological effects as priming and habituation could be reversed. They explored whether participants could respond to a priming word before being exposed to said word – that is, whether people could be retroactively primed.

Unsurprisingly, precognition, premonition, and other psychic phenomena (collectively known as "psi" phenomena) are controversial topics and are frequently the subject of heated debate. However, Bem's original nine experiments (2011) spurred a level of controversy unusual even among psi research, for several reasons. First, Daryl Bem is a well-known researcher; he proposed the self-perception theory of attitude change (the idea that people can infer their attitudes from their own behavior as outside observers might; if someone gave a speech that was pro-Fidel Castro, for instance, they would be more likely to perceive themselves as in favor of Castro) (Bem, 1971). Bem is also known for his "Exotic Becomes Erotic" (EBE) theory (Bem, 2000), which posits that children who are attracted to activities enjoyed by children of the opposite gender are more likely to view members of their own gender as exotic and, therefore, erotic later in life. But Bem's psi experiments did not attract attention only because of Bem's history. The study containing Bem's initial nine experiments (2011) was published in a reputable journal, the *Journal of Personality and Social Psychology* (*JPSP*). *JPSP* has a high impact factor (4.736 in 2015) and subjects its articles to rigorous peer review; Bem himself served for a time as one of its associate editors. In publishing his experiments,

Bem intended to produce evidence that even skeptics could not ignore – and, after a thousand total subjects and an investigation that spanned ten years, he succeeded.

Once Bem's experiments were published, they became a media phenomenon, featured in such venues as the front page of the *New York Times*, *The Colbert Report*, and online blogs like *Wired*. This was likely more attention than any single previous psi study had ever attained, and such a flood of attention brought with it a flood of replications. To his credit, Bem expected and even ardently encouraged these replications; he intentionally kept his methodology and statistical analyses simple and provided extensive instructions. Of this flood of replications, Engber of *Slate* (2017) wrote,

> "If one had to choose a single moment that set off the 'replication crisis' in psychology – an event that nudged the discipline into its present and anarchic state, where even textbook findings have been cast in doubt – this might be it: the publication, in early 2011, of Daryl Bem's experiments on second sight."

The dataset featured in this dissertation (Bem, Tressoldi, Rabeyron, & Duggan, 2016) includes 69 of these replications, along with the original studies, and assesses the overall effect of retroactive stimuli.

Of course, if a researcher is to demonstrate the impact of study characteristics on publication, relevant characteristics must be present. It is possible to examine the characteristic of "authorship," denoted as "Bem" vs. "other" – that is, do the nine studies published by Bem demonstrate a different pattern of publication? Nine studies, however, is a very small number in one level of the moderator, which makes this moderator a poor candidate. Another (and perhaps a more interesting) question involves the year of publication. The meta-analysis includes 30 studies conducted pre-2011, before Bem's research exploded onto the field, and 39 studies conducted post-2011 (excluding those published *in* 2011). If Bem's research truly did spark a "replication crisis," is such a crisis reflected in publication patterns? One wonders whether publication bias may have drastically decreased after 2011, as researchers scrambled to publish all replication results … including the unfavorable ones.

2.1.2 Analyses

I use *R* version 3.4.2 (*R* Core Team, 2017) and its graphical user interface (GUI) *RStudio*, version 1.1.383 (*R* Core Team, 2017), along with the *R* packages *metafor* (Viechtbauer, 2010), *ggplot2* (Wickham, 2009), and *weightr* (Coburn & Vevea, 2017), to conduct the example analyses throughout this dissertation.

Figure 1 and Figure 2 present funnel plots of the effect sizes from studies published before 2011 and after 2011, respectively. On each plot, a horizontal line marks the unadjusted mean. Recall the guidelines for interpreting funnel plots that were presented in Chapter 1 of this dissertation. If no relationship exists between study size, or standard error, and effect size, the plot should give the impression of a horizontal funnel around the mean. Asymmetry in the plot indicates that there may be a relationship between study size and effect size, or that publication bias may be present (with the caveat, of course, that moderators of effect size can also give the impression of asymmetry).

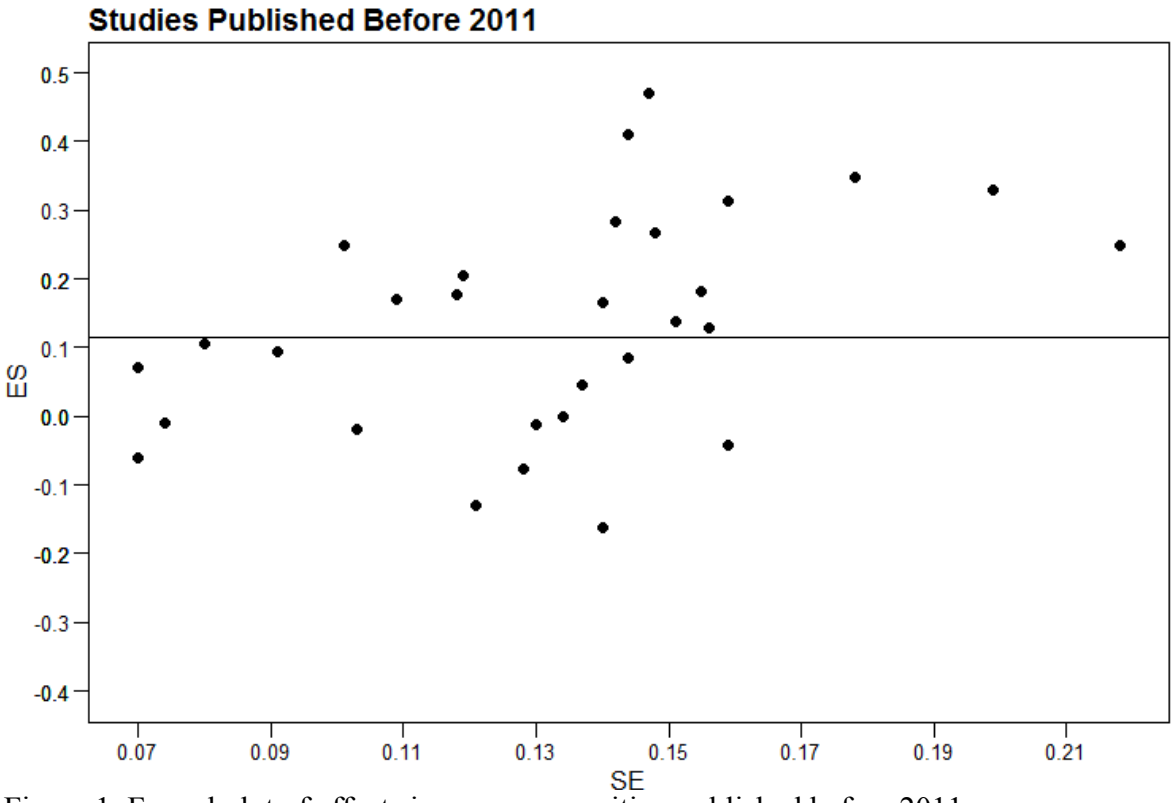


Figure 1. Funnel plot of effect sizes on precognition published before 2011.

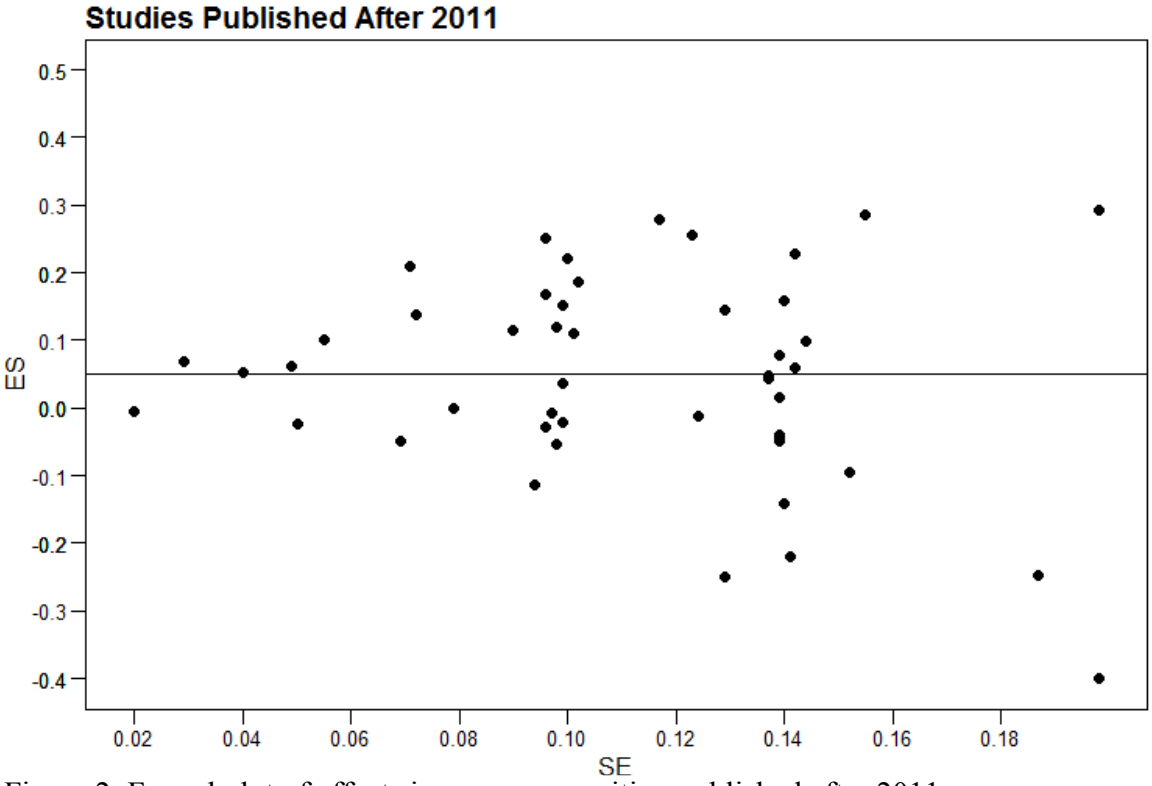


Figure 2. Funnel plot of effect sizes on precognition published after 2011.

Figure 1, of the earlier studies, is less than ideal. There appear to be more positive effect sizes published, particularly in the region of about 0.1 to 0.5, than negative ones. As a result, the unadjusted mean is pulled upward – note that the horizontal line does not appear to be in the middle of the cloud of points and does not correspond to the apparent apex of the "funnel" at the left side of the plot.

In contrast, funnel plots rarely appear more symmetric than Figure 2. The horizontal funnel is clear. There is little to no visible asymmetry. As a result, the horizontal line almost perfectly bisects the cloud of points.

These funnel plots are a preliminary investigation, of course. They are not a significance test for the presence of bias, nor can they produce an adjusted mean or variance-component estimate for either group. However, the fact that these analyses are rudimentary does not diminish their worth.

There is clearly asymmetry in the plot of the earlier studies that is not remotely present in the later studies. This piece of evidence supports the theory that publication bias, in the field of precognition, may have drastically decreased after 2011. At the very least, the pattern of publication certainly seems to have changed. I will continue to explore this substantive dataset with the models described in the following chapters.

2.2 Conclusions

This chapter describes some study characteristics that can operate as moderators of publication bias. It then presents an empirical dataset, obtained from a published meta-analysis, that is an example of such phenomena, and demonstrates that symmetry in

funnel plots may visibly differ across levels of a study characteristic (here, year of publication).

The next chapter, Chapter 3, describes the parameters of a simulation that I will refer to throughout the dissertation to explore the performance of the lambda model and its variants. Description of the model begins in Chapter 4.

Chapter 3: Simulation Design

To assess the performance of the models presented in this dissertation, I conducted an extensive simulation. Rather than repeating the description of its design once per chapter, I use this chapter to outline the simulation structure; the results of the simulation for each variant of the model are then presented in sections at the ends of the relevant chapters.

The lambda model and its variants are new, so it is important to assess their performance under ideal circumstances – that is, in cases where selection bias is generated according to the model, where the sample size (or the number of effect sizes in the meta-analysis) is large, and where between-studies heterogeneity (or the variance component) is small. These circumstances are where the models are likely to perform the best. Once their baseline performance is established, it is useful to know how the models perform when their assumptions are violated. To that end, I varied the size of the variance component, the number of studies, the method of generating selection bias, and the pattern of bias. The lambda models, of course, will not perform as well when bias is generated according to a different model, but it is crucial to explore their performance under such conditions because it is impossible to know the true generating model for publication bias.

All data were generated using *R* version 3.4.2, "Short Summer" (R Core Team, 2017). I also used several *R* packages, namely *weightr* (Coburn & Vevea, 2015), *R2jags* (Su & Yajima, 2015), *tictoc* (Izrailev, 2014), *metafor* (Viechtbauer, 2017), and *doParallel* (Calaway, 2017). To implement the Bayesian models, I used the free software Just Another Gibbs Sampler, or JAGS (Plummer, 2017), which can be called via *R* through packages like *R2jags* (Su & Yajima, 2015). The entire simulation code is provided in Appendix B. Data were generated in the form of standardized mean difference effect sizes (d), with a true population mean of $d = 0.20$.

I varied four factors in this simulation – the number of effect sizes (or studies) per meta-analytic model (k), the percentage of between-studies heterogeneity (measured in terms of I^2), the pattern of selection bias, and the method of bias generation. There are four levels of each factor, resulting in a total of 4 x 4 x 4 x 4 = 256 simulation cells. The factors and their corresponding levels are discussed in more detail below.

The levels of the first factor, number of studies, are $k = 12$, 24, 48, and 172. These levels are based on previous work in which I, along with an undergraduate research assistant, surveyed the number of studies (k) included in empirical social science and medical meta-analyses (Coburn, Vevea, & Orey, in prep; Coburn & Vevea, in prep). We searched *Psychological Bulletin*, the premier journal for meta-analyses in psychology, and the *Campbell Collaboration*, the social sciences database of systematic reviews, to represent the social sciences; for their medical counterparts, we searched the *British Medical Journal (BMJ)*, a journal that publishes a relatively large number of meta-analyses, and the *Cochrane Collaboration*, the medical database of systematic reviews (Coburn & Vevea, in prep). Table 1 presents an overview of the descriptive statistics of these four distributions.

Table 1. Descriptive statistics of distributions of the number of studies (k), trimmed.

Estimate	BMJ	Cochrane	Psych Bull	Campbell
Mean	57.76	13.02	119.22	27.82
Standard deviation	119.25	18.68	118.88	38.50
Minimum	0.00	0.00	6.00	0.00
Maximum	965.00	260.00	919.00	265.00
Quantiles				
0.05	6.00	0.00	22.00	0.00
0.10	7.00	1.00	30.50	1.60
0.25	11.00	3.00	47.25	6.00
0.50	21.00	7.00	84.50	13.00
0.75	46.00	16.00	151.50	35.00
0.90	122.00	30.50	238.50	61.00
0.95	259.00	47.25	292.25	78.40

Largest *k* trimmed from Psych Bull. Largest two *k*s trimmed from Cochrane.

Overall, Table 1 demonstrates that medical meta-analyses, such as those published in the *BMJ* and the *Cochrane Collaboration*, tend to be quite small, with a 50th percentile of $k = 21$ and $k = 7$, respectively. On the other hand, social science meta-analyses can be large, with a 50th percentile of $k = 84.5$ for *Psychological Bulletin* (Coburn & Vevea, in prep; Coburn, Vevea, & Orey, in prep). I selected levels of *k* that represent both the high and low ends of these distributions; I also ensured that the levels were evenly divisible by two, as the models described in this dissertation incorporate a dichotomous moderator. *k* of 12 is about the 25th percentile of the *BMJ*; *k* of 24 is about the 5th percentile of *Psychological Bulletin*; *k* of 48 is about the 75th percentile of the *BMJ* and the 25th percentile of *Psychological Bulletin*; and finally, *k* of 172 is about the 90th percentile of the *BMJ* and the 75th percentile of *Psychological Bulletin*. The number of effect sizes in each level of the dichotomous group membership moderator per level is 6, 12, 24, and 86 (the total number divided by two).

For the second factor, the percentage of between-studies heterogeneity, the four levels each correspond to values of I^2. The I^2 index is based on the ratio of between-studies heterogeneity to total heterogeneity (including sampling variance), and this ratio is expressed as a percentage. The four levels represent I^2 of 0%, 25%, 50%, and 75%. These levels are based on a tentative classification of I^2 by Higgins and Thompson (2002), who proposed that values of 25%, 50%, and 75% represent low, medium, and high heterogeneity, respectively (Coburn & Vevea, in prep). However, it is impossible to generate effect sizes based solely on a value of I^2; as described above, it is a ratio. Therefore, it is necessary to transform I^2 into a variance component, or τ^2. The relationship between I^2 and τ^2 can be generally described as follows:

$$I^2 = \frac{\tau^2}{(\tau^2 + \sigma^2)}$$

Or, rewritten to solve for τ^2:

$$\tau^2 = \frac{I^2\sigma^2}{(I^2 - 1)}$$

This formula may clarify the description of I^2; that is, the ratio between the amount of between-studies heterogeneity, τ^2, and the total amount of heterogeneity, including sampling variance, $(\tau^2 + \sigma^2)$. This simplified formula, however, operates under the assumption that the sampling variance is fixed across effect sizes, or that each effect size in a given meta-analysis has the same sampling variance (i.e., the same sample size). This situation is *extremely* unlikely to occur in practice. Therefore, the researcher must choose between sacrificing realism for the sake of accuracy and generating fixed sampling variances or sacrificing a degree of accuracy and permitting the sampling variances to differ (Coburn & Vevea, in prep). I chose the latter and worked with a database of empirical meta-analytic study sample sizes to obtain an average sampling variance (σ^2) from which to generate population values of τ^2.

To maintain empirical validity, or to emulate situations that are likely to arise in practice, my lab obtained a dataset of the sample sizes for individual studies from meta-analyses in several of the primary branches of psychology (namely, industrial/organizational, health, developmental, social, and clinical). These sample sizes are from meta-analyses of standardized mean difference effect sizes, or *d*-statistics. The resulting distributions of three fields (industrial/organizational, social, and health) contained several studies with extremely large sample sizes (for example, 4,276), while the other two distributions (clinical and developmental) did not (Vevea, Zelinsky, Turitz Mitchell, Castaneda, & Coburn, in prep; Coburn & Vevea, in prep). After surveying these distributions, we wrote two *R* functions to generate any specified number of sample sizes with very similar distributions (Vevea et al., in prep; Coburn & Vevea, in prep). The first of these functions, modeled after the fields of industrial/organizational, social, and health, contains several extremely large sample sizes; the other, modeled after clinical and developmental, does not. For this simulation, I used the latter to generate a large number of sample sizes (k = 10,000) and computed their sampling variances. I then used the median of this semi-empirical distribution, 0.08, as an estimate of σ^2. Some algebra yielded τ^2 values of 0.00, 0.03, 0.08, and 0.23, which correspond to I^2 values of 0%, 25%, 50%, and 75%, respectively. This approach will not be perfectly accurate but should be an acceptable compromise.

The third simulation factor is the pattern of selection bias. For each method of bias generation, I create "strong" and "weak" examples of bias. In this simulation, effect sizes are divided into two groups, and the degree of publication bias varies across groups. The levels of this third factor are then: (1) "None vs. None"; (2) "None vs. Weak"; (3) "None vs. Strong"; and (4) "Weak vs. Strong." In the first level, no bias is present in either group; in the second, no bias is present in one group and weak bias in the other; and so on. This factor is important because it will demonstrate whether this model results in biased estimates when there is no bias in either group (the first level). It will also indicate whether the model reproduces estimates more accurately when, for instance, weak bias is present rather than strong bias. The fourth level of this factor will indicate how well the model can adjust for cases in which both groups suffer from bias, but to differing degrees.

Finally, we arrive at the fourth factor, the method of bias generation. This factor is a bit more complicated. There are four levels, described here separately. Each of these levels represents a different way of violating model assumptions; they are not intended to represent actual proposed selection mechanisms. For each level, I demonstrate conditions of what I dub "strong" and "weak" publication bias. Note that it is not necessary to demonstrate the conditions with no publication bias, as all effect sizes are simply retained.

3.1 Level One: Step Function of *p*-value

For these cells, selection bias is generated exactly according to the model assumptions. The Vevea and Hedges (1995) model and, consequently, the lambda model both represent selection bias using weighted distribution theory, in which a weight function describes the likelihood of effect sizes in a given range (e.g., $p < .05$) being observed (Vevea & Hedges, 1995; Iyengar & Greenhouse, 1988; Hedges, 1984; Lane & Dunlap, 1978). It is possible to specify a variety of weight functions, of course. The Vevea and Hedges (1995) model, and its counterparts, use a step function based on user-specified psychologically relevant *p*-value cutpoints, such as $p = .01$, $p = .05$, $p = .10$, $p = .50$, and so on. The model uses a step function because the likelihood of survival is likely to be drastically higher for a study with a *p*-value of .049 than for a study with *p* of .051, for example, and because there is likely to be little to no difference in the likelihood of survival within the same *p*-value interval. The latter means that studies with *p* of .051 and *p* of .059, for instance, are equally likely, or unlikely, to survive. For an illustration of this type of step function, see Figure 3.

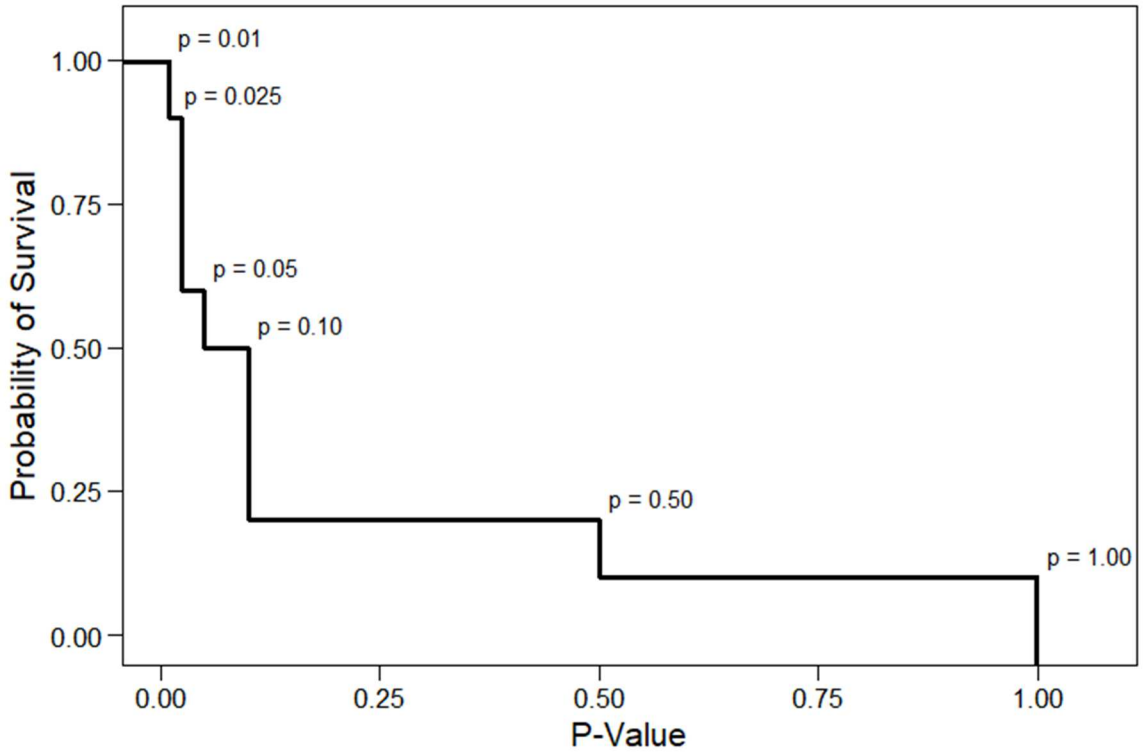


Figure 3. An illustration of a step function based on *p*-values.

I generated selection bias according to this model by simulating effect sizes and retaining given proportions of effect sizes within specified *p*-value intervals. It is possible to specify any number of *p*-value intervals, but for this simulation I used a simplified approach distinguishing only between significant and nonsignificant effect sizes, with one cutpoint at $p = .05$.

For "strong" selection bias, only 20% of nonsignificant effects survive, while 100% of significant ones do. Figure 4 illustrates its step function.

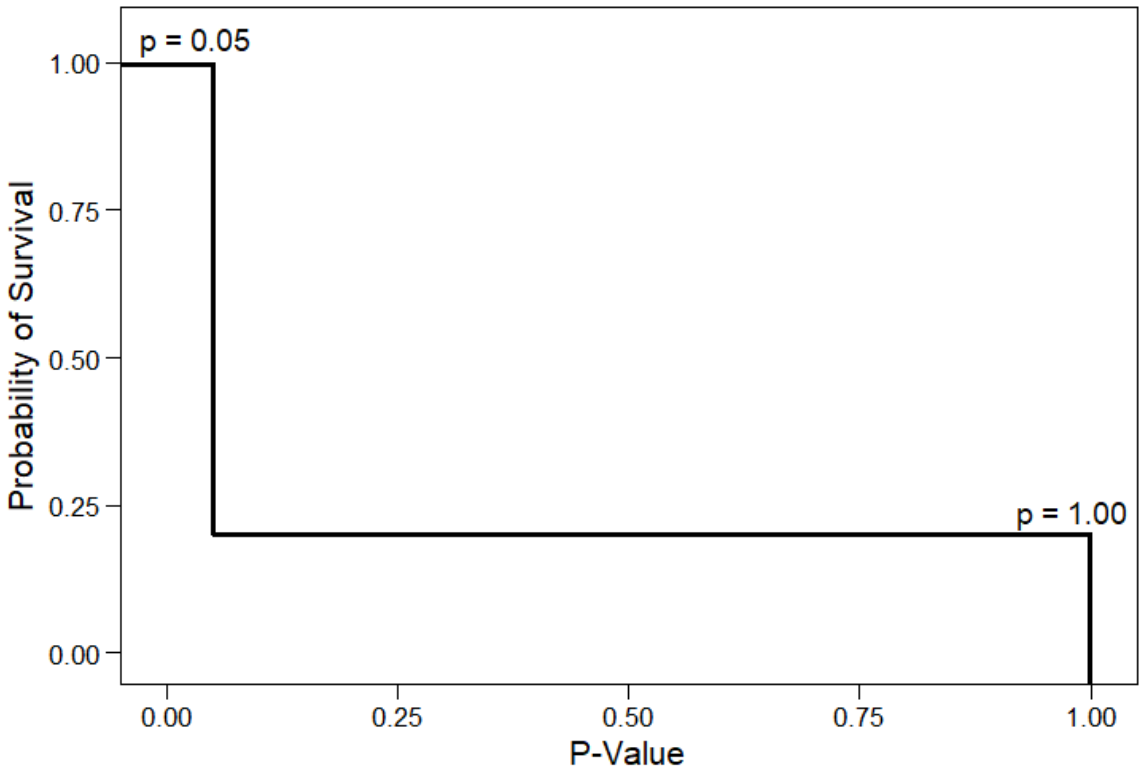


Figure 4. A step function based on *p*-values, representing strong publication bias.

For "weak" selection bias, 70% of nonsignificant effects survive, while 100% of significant ones do. Figure 5 illustrates its step function.

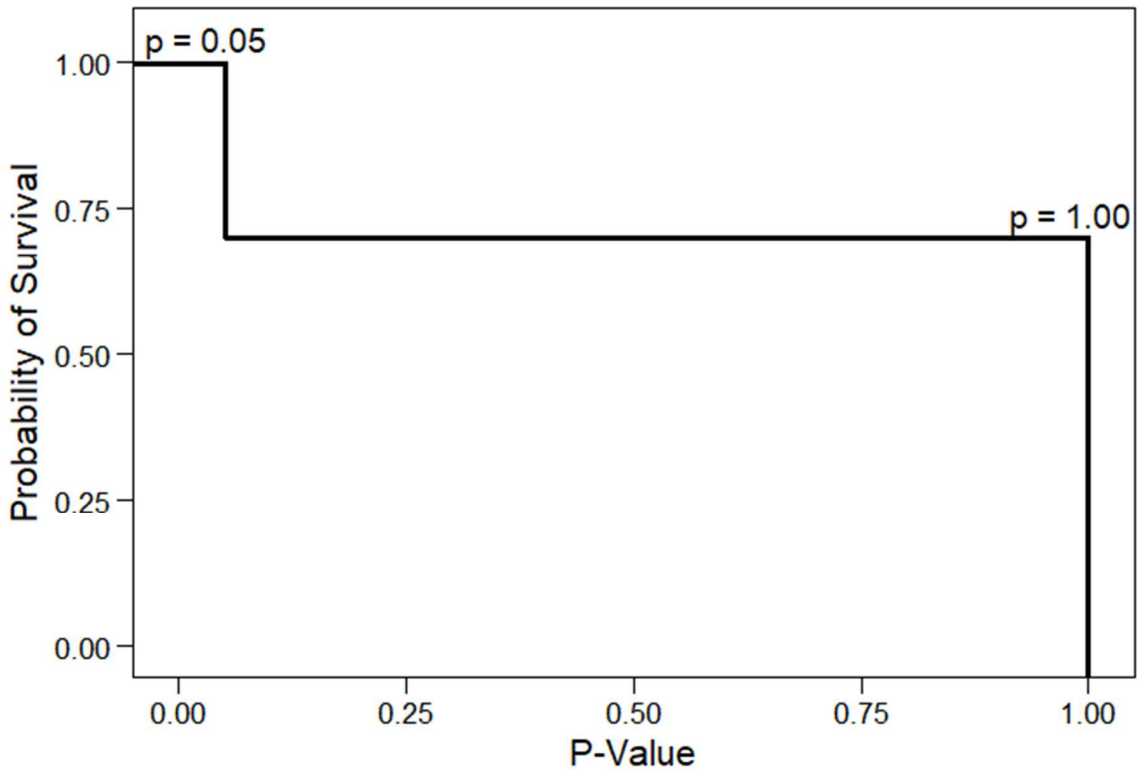


Figure 5. A step function based on *p*-values, representing weak publication bias.

The lambda model and its variants will likely perform best when bias is generated according to the model assumptions. It is also worthwhile to note that a step function with psychologically relevant *p*-value cutpoints is likely to be one of the most representative and empirically valid patterns of publication bias. However, it is always possible that existing bias is generated by some other mechanism, so it is important to explore the model's performance under varying methods of bias generation.

3.2 Level Two: Exponential Function of *p*-value

For Method Two, I generated selection bias as an exponential function of *p*-values. The underlying idea is the same – that an observed effect size's chance of survival, or of publication, is based on its corresponding *p*-value. However, the first method used a step function, which assumes that there is a steep drop-off in the chance of survival after given *p*-value cutpoints. This method uses an exponential function, assuming that the chance of survival decreases at a rate proportional to its current value.

The general formula for an exponential function is as follows:

$$f(x) = e^{x}$$

It is also possible to add a "rate" parameter, ζ:[3]

[3] λ is standard notation for the rate parameter. My use of ζ is non-standard, but avoids conflict with the λ parameter in the lambda model.

$$f(x) = e^{(-\zeta * x)}$$

Exponential distributions are often used in the context of change over time, such as compound interest. In a time context, the rate parameter describes the expected number of events within a given time interval. In a survival context, however, the rate parameter is a kind of "hazard" estimate, or an estimate of constant risk. Increasing the rate parameter value increases the size of the number being exponentiated, and therefore increases the function's rate of change. This results in a steeper exponential curve, or stronger publication bias. Decreasing the rate parameter value has the opposite effect.

For "strong" selection bias, I calculated the probability of survival as follows, with a rate parameter of 2. Figure 6 plots the corresponding function.

$$prob = e^{(-2 * p-value)}$$

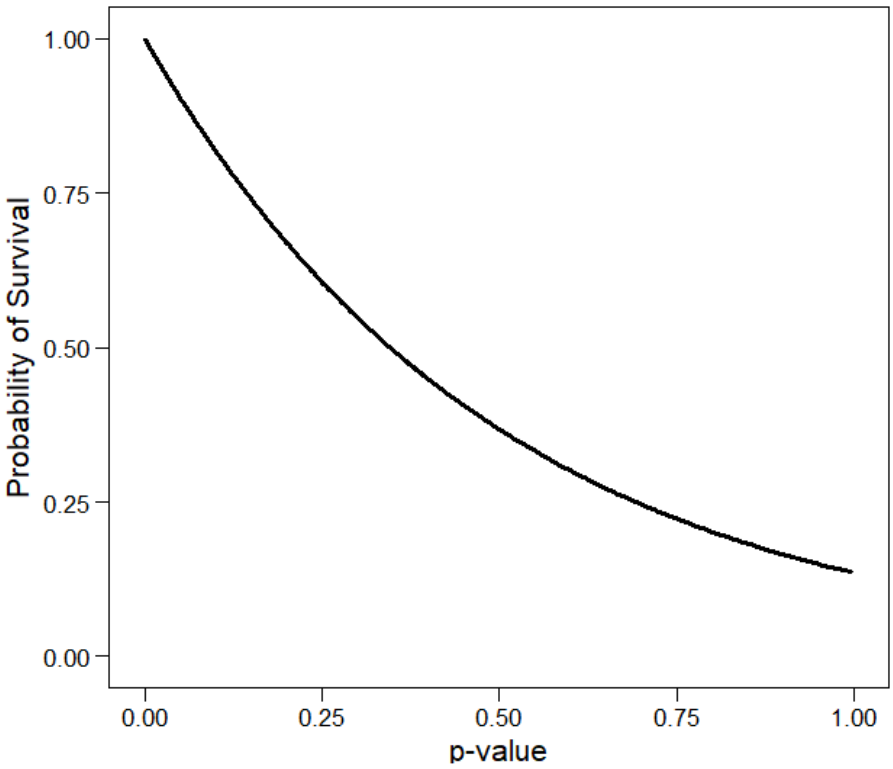


Figure 6. An exponential function based on *p*-values, representing strong publication bias.

For "weak" selection bias, I calculated the probability of survival as follows, with a rate parameter of 0.50. Figure 7 plots the corresponding function.

$$prob = e^{(-0.5 * p-value)}$$

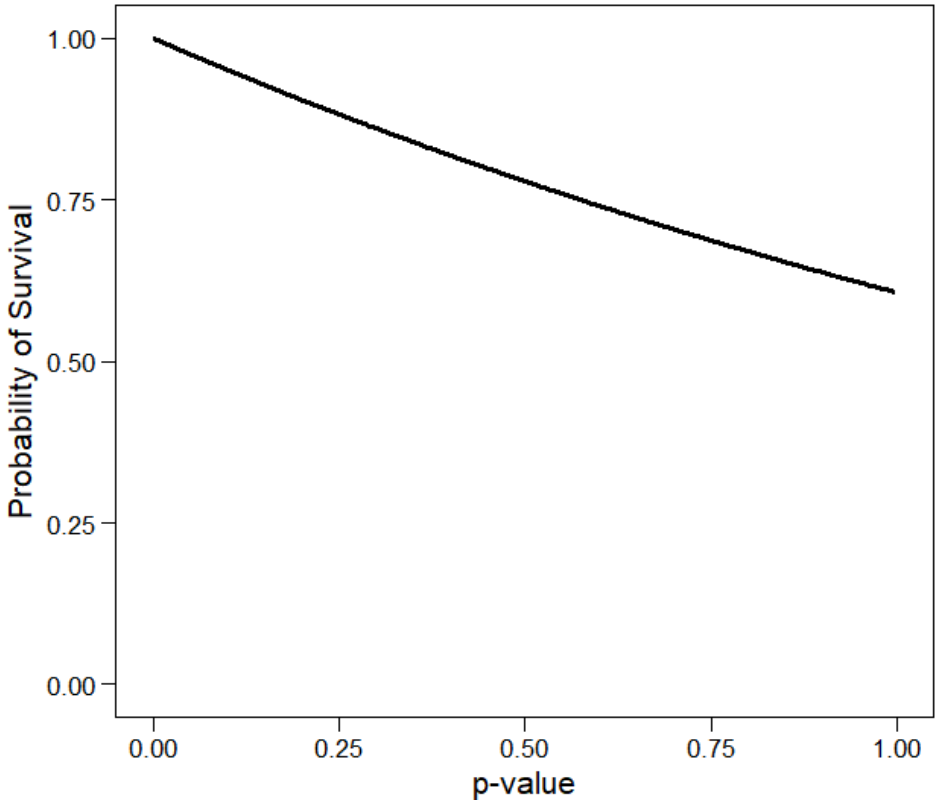


Figure 7. An exponential function based on *p*-values, representing weak publication bias.

When "strong" bias is present, *p*-values close to zero have essentially a 100% chance of survival, and those close to one have less than a 25% chance. When "weak" bias is present, on the other hand, *p*-values close to one still retain about a 60% chance of survival.

3.3 Level Three: Step Function of Effect Size

In certain fields, like the area of single-case design, the likelihood of publication can depend on factors other than *p*-value, such as effect size. (When it comes to single-case design research, for instance, *p*-values are typically not even calculated.) For this level, I generate selection bias using a step function, very much like Model One; the only difference is that the cutpoints are based on an effect-size metric rather than a *p*-value metric.

The most commonly-cited guidelines for assessing the magnitude of *d*-statistics were proposed by Cohen (1988). For standardized mean difference effect sizes, like the *d*-statistics generated here, Cohen defined small, medium, and large effect sizes as $d =$ 0.20, 0.50, and 0.80, respectively. Cohen proposed these guidelines under the very specific context of selecting population effect sizes for power analysis (Cohen, 1988), but the use of them as thresholds for substantive interpretation has spread across the research universe like a plague. In 1977, Cohen noted that "this is an operation fraught with many dangers," and in 1988 he warned readers about the importance of being flexible with these values, specifically avoiding their use as de facto standards (Cohen, 1988; Lenth,

2001). Much like a worldwide game of Telephone, though, his message has been garbled beyond recognition, and today one is hard-pressed to find a researcher who knows the true meaning of Cohen's guidelines.

I specified one effect-size cutpoint at *d* of 0.50. This value corresponds to Cohen's "medium" effect size. The value *d*=0.50 is likely to be a psychologically relevant effect size, given the popularity of Cohen's guidelines – such that larger effect sizes are more likely to be published, and smaller ones less likely. This structure also mirrors the structure of Method One.

I emphasize that my use of this cutpoint absolutely should not indicate that I support the use of Cohen's guidelines in any context outside that of power analysis. However, I am aware that most of the research community do base their interpretations on his guidelines, and therefore a step function based on this cutpoint is psychologically relevant.

For "strong" selection bias, 100% of effect sizes above 0.50 survive, while only 20% of those below do. Figure 8 plots the corresponding step function.

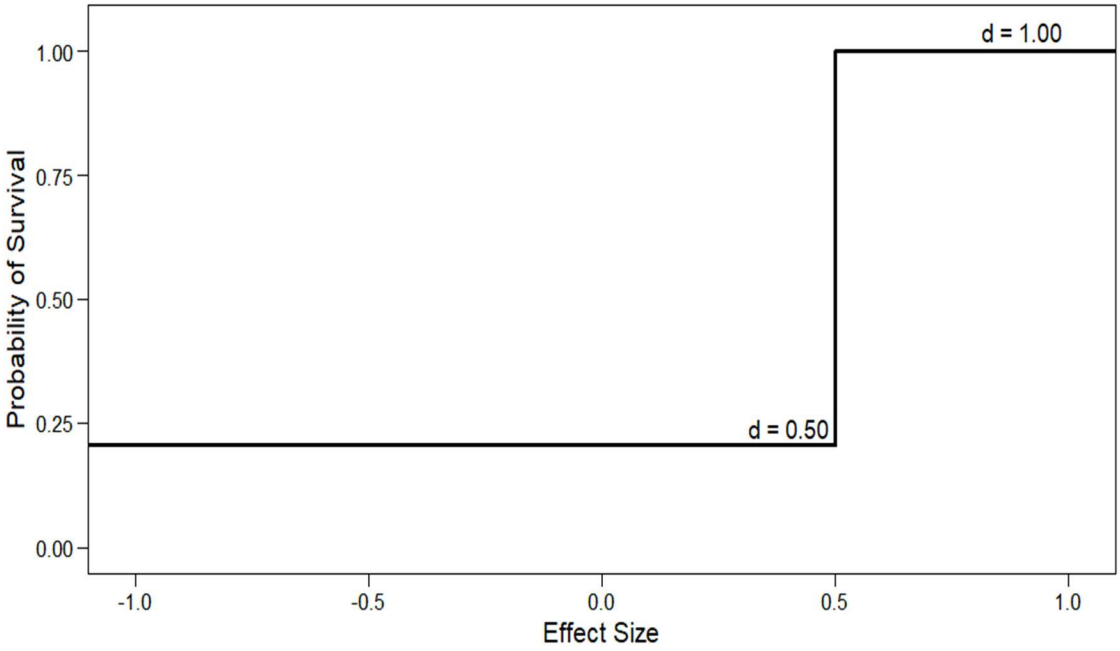


Figure 8. A step function based on effect sizes, representing strong publication bias.

For "weak" selection bias, 100% of effect sizes above 0.50 survive, compared to 70% of those below. Figure 9 plots the corresponding step function.

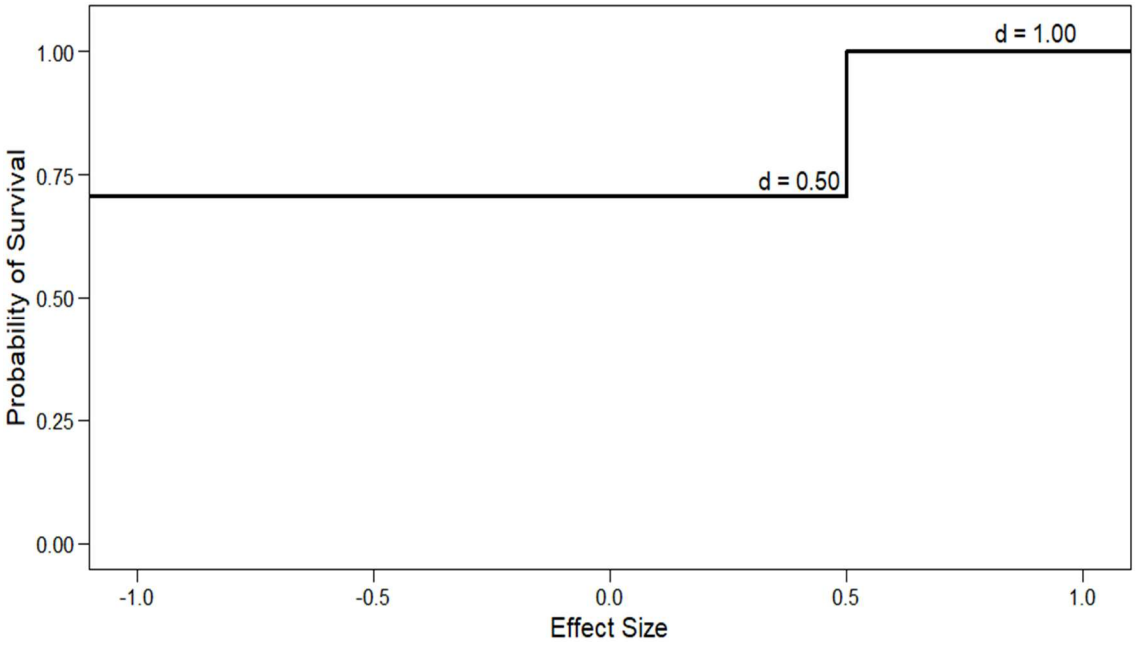


Figure 9. A step function based on effect sizes, representing weak publication bias.

3.4 Level Four: Logistic Function of Effect Size

For consistency, and to parallel Level Two the way that Level Three parallels Level One, it makes sense to generate selection bias as an exponential function of effect size. However, doing so poses a problem. Unlike *p*-values, effect sizes can be negative numbers. It is still possible to exponentiate a negative number, of course. The problem is that doing so can yield negative values of $f(x)$. In this situation, $f(x)$ represents effect sizes' probability of survival, and a probability cannot be a negative number. One could simply remove those effect sizes with negative probabilities, but artificially trimming the distribution in such a way would contribute additional bias.

To generate selection bias based on effect size, therefore, I used a logistic function, which must yield $f(x)$ values between zero and one. More specifically, I used the two-parameter logistic (or 2PL) item response theory (IRT) model. IRT is a statistical paradigm for testing participants, or measuring participants, regarding some latent trait of interest, often denoted as theta (Θ). In such a testing scenario, the probability of responding correctly to a dichotomous item *i* (usually a multiple-choice question) is a function of several parameters, including the individual's Θ value. In this case, I am modeling publication bias, so the probability of survival (which is dichotomous – an effect either survives or fails to survive) is a function of several parameters, including the individual effect size, Θ.

In these equations, note that subscripts are omitted for the sake of simplicity.

The standard logistic function is as follows:

$$prob = \frac{e^{\theta}}{1 + e^{\theta}}$$

The two-parameter logistic function (2PL) is:

$$prob = \frac{e^{a*(\theta - b)}}{1 + e^{a*(\theta - b)}}$$

The standard logistic function yields a sigmoid curve. The additional parameters determine various aspects of the shape and location of the curve. The *a* parameter represents the maximum slope of the curve, or its steepness (in IRT, this is referred to as the *discrimination* parameter). The *b* parameter represents the location on the *x*-axis where the curve is centered (in IRT, this is the *difficulty* parameter). Three-parameter (3PL) and even four-parameter functions also exist, in which asymptotic minima or maxima can be specified; however, including such parameters is not necessary here, as the outcome is naturally bounded between zero and one.

For strong selection bias, I set the *a* parameter to 5 and the *b* parameter to 0.464, resulting in:

$$prob = \frac{e^{5*(\theta - 0.464)}}{1 + e^{5*(\theta - 0.464)}}$$

The corresponding logistic function is illustrated in Figure 10.

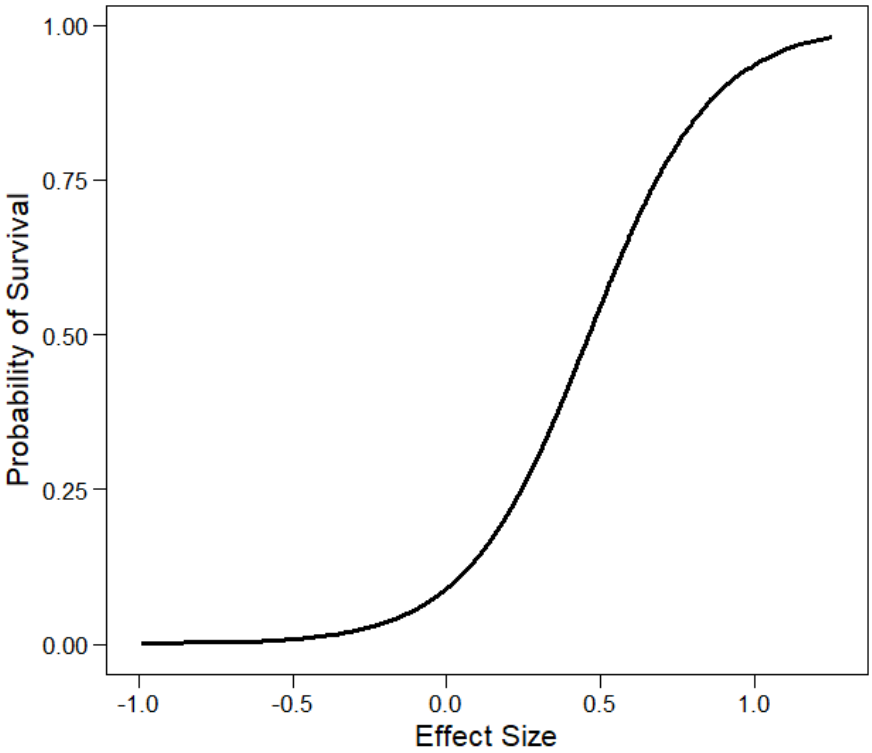


Figure 10. A logistic function based on effect sizes, representing strong publication bias.

The resulting curve, shown above, is centered at $d = 0.464$. I chose this center point so that a change in selection occurs approximately around p of 0.05. The curve is steep, with a maximum slope of 5.

For weak selection bias, I reduced the value of the a parameter to lessen the steepness of the curve, setting a to 3 and b to -0.464:

$$prob = \frac{e^{3*(\theta-(-0.464))}}{1 + e^{3*(\theta-(-0.464))}}$$

The corresponding logistic function is illustrated in Figure 11.

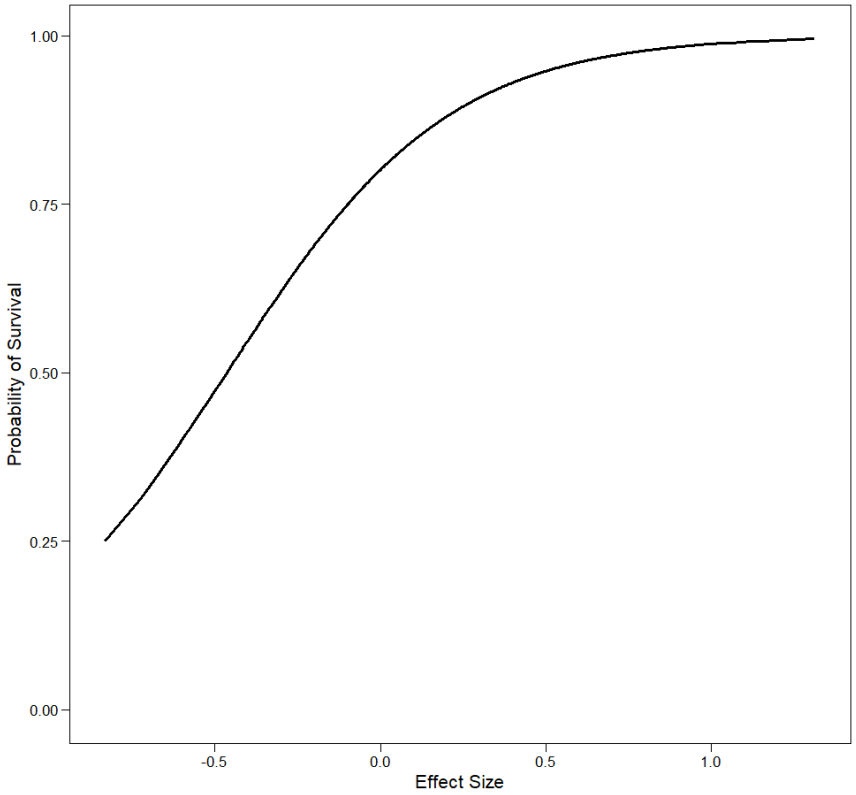


Figure 11. A logistic function based on effect sizes, representing weak publication bias.

This curve, featured above, is much less steep. It is centered at $d = -0.484$, corresponding to a p-value of 0.95, or representing one-tailed significance in the opposite direction. This mirrors the strong selection bias example and ensures that the chance of survival is higher for the negative effect size values, representing a weaker pattern of publication bias. A maximum slope of 3 ensures that the curve is fairly shallow.

When "strong" bias is present, effect sizes below d of 0.464 have a drastically reduced chance of survival, decreasing to about 0% for effects below d of -0.50 and increasing to about 100% for effects above d of 0.90. When "weak" bias is present, on the other hand, effect sizes below d of -0.50 still have an almost 50% chance of survival, and effect sizes above d of 0.464 are about 100% likely to survive. In this way, although bias based on effect size is still present in the weak condition, it is much less severe.

This simulation design results in a total of four factors, each with four levels, or a total of 256 independent simulation cells. However, keep in mind that the fourth factor, the pattern of selection bias, contains the level "None vs. None," which compares two groups without any selection bias. No selection bias is present, so, for this level, all effect sizes are retained. Therefore, regardless of the method of bias generation, if no bias is present, there is no need to re-analyze these cells across methods. This results in a total of 208 independent simulation cells.

The rest of this dissertation explores the creation and evaluation of a weight-function model that is capable not only of detecting the presence of differing publication bias patterns but also of conducting a significance test, including moderators of effect size, and providing adjusted estimates of all parameters. This model is henceforth known as the lambda model.

Chapter 4: The Lambda Model

Although our previous paper (Coburn and Vevea, 2015) presents some introductory approaches to the problem, no formal model currently exists that can assess moderators of publication bias. Therefore, this dissertation develops one.

The ideal model for this situation is one that can incorporate both random and systematic sources of heterogeneity while requiring minimal additional parameters to describe the publication bias process. This ideal model would yield adjusted estimates of the intercept, the variance component, and any moderator variables, and these estimates should maximize accuracy and power while minimizing bias and Type I error (qualifications for any ideal estimator). The model should also produce information about the pattern of publication bias and, if possible, information about the impact of study characteristic(s) on said pattern.

It is often more efficient to modify a pre-existing model, rather than developing a model which meets these requirements from scratch. The Vevea and Hedges (1995) weight-function model, as described in Chapter 1, is likely to be the best candidate. It can accommodate both random and systematic heterogeneity, it yields adjusted estimates for all parameters, and it produces some information about the pattern of bias (through its estimates of the weights for each *p*-value interval). Although selection models are sometimes dismissed because they can require large numbers of effect size or due to their computational intensity, my modified version of the Vevea and Hedges (1995) model can circumvent these arguments. Chapter 5 presents a simplified version of the lambda model which incorporates fixed weights to reduce the number of effect sizes required. Bayesian implementation, rather than maximum likelihood, also requires fewer effect sizes due to the inclusion of prior distributions; this version of the model is presented in Chapter 6. As for computational intensity, the *R* package *weightr* (described in Chapter 7) allows empirical meta-analysts to use the Vevea and Hedges (1995) model and the lambda model with one line of code in the free, multi-platform, open-source statistical software *R* (R Source Team, 2013).[4]

An adapted version of the Vevea and Hedges (1995) model must allow the pattern of bias to vary across levels of a moderator; it must do so with as few additional parameters as possible; and it must possess good statistical qualities. Both Coburn and Vevea (2015) and Coburn and Vevea (in prep) use the Vevea and Hedges (1995) model to assess publication bias across levels of a moderator variable, but they do so simply by estimating the model once per level. This procedure doubles the number of weight parameters estimated and requires many effect sizes both per interval and per level. Doing so technically allows the pattern of bias to vary, but it does not do so practically; estimating the model once per level means that the meta-analyst is very restricted in terms of how many *p*-value weights can be estimated, which in turn limits the information about the selection-bias pattern. This problem means that, although estimating the model once per level is technically possible, it is neither practical nor

[4] The lambda model feature of *weightr*, although described here, will not be publicly available until the relevant research is published.

plausible. In contrast, the lambda model solves the problems described above, and it does so without requiring a larger number of effect sizes and while allowing the mean and variance component to remain constant across levels. Therefore, it may be a workable solution.

The lambda model is based on the original Vevea and Hedges (1995) model, which functions as described in Chapter 1. The lambda model incorporates study characteristics as moderators of the weights for the nonsignificant *p*-value intervals. In cases where the relevant study characteristic is a continuous variable, the procedure is straightforward. In cases where the study characteristic is categorical (e.g., separating studies that were published before and after an event), the variable can be dummy-coded. If there are two categories, or levels, of the study characteristic, there need only be one dummy-coded variable; effect sizes are then coded 0 if they belong to one level and 1 if they belong to the other. In this case, the lambda model estimates only one additional parameter, which I refer to as lambda (λ). λ contains information about the difference in the selection-bias pattern for the group coded 1. λ is a multiplicative constant on the weights for the nonsignificant *p*-value intervals. The model can estimate a full set of weights for one group and provides information about the second group through the estimation of λ. In principle, as described above, meta-analysts could specify a continuous study characteristic, a categorical characteristic with more than two levels, or multiple study characteristics; however, the version of the model presented in this dissertation employs one dummy-coded variable.

Much like the Vevea and Hedges (1995) model, the lambda model includes both an effect-size component and a selection component. The effect-size component is a fixed-effect, random-effects, or mixed-effects meta-analytic model, like those described in Chapter 1. The selection model, which illustrates the relative likelihood of survival for effect-size estimates with particular *p*-values, differs. The selection model is defined as the weighted probability density function of Y_i (the effect sizes), given parameters $\boldsymbol{\beta}$, τ^2, $\boldsymbol{\omega}$, and λ:

$$f(Y_i|\boldsymbol{\beta},\tau^2,\boldsymbol{\omega},\lambda;\sigma_i^2,Z_i,\boldsymbol{X_i}) = \frac{1}{\sqrt{\sigma_i^2+\tau^2}} \frac{w(Y_i,\sigma_i^2,Z_i)\phi\left(\frac{Y_i-\Delta_i}{\sqrt{\sigma_i^2+\tau^2}}\right)}{\int_{-\infty}^{\infty}\frac{1}{\sqrt{\sigma_i^2+\tau^2}}w(Y_i,\sigma_i^2,Z_i)\phi\left(\frac{Y_i-\Delta_i}{\sqrt{\sigma_i^2+\tau^2}}\right)dY_i}$$

where λ is a multiplicative constant on certain nonsignificant weights (see below), τ^2 is the between-studies variance component, Y_i and σ_i^2 denote the effect sizes and their corresponding sampling variances respectively, $\phi(z)$ the standard normal probability density function evaluated at z, and Δ_i represents $\boldsymbol{X_i\beta}$. In a mixed-effects model, $\boldsymbol{\beta}$ is a q-dimensional vector of unknown regression coefficients, $(\beta_0, \beta_1, \ldots, \beta_q)$, and X_i is a vector of known predictors, $(X_{i1}, \ldots, X_{iq})'$. In a fixed-effect or random-effects model, q is equal to one, representing the intercept (or the overall mean). Multiplying the two ($\boldsymbol{X_i\beta}$)

yields a function of linear predictors, allowing the meta-analyst to estimate models with varying numbers and types of predictors. When only the intercept is estimated, Δ_i represents a mean-only fixed-effect or random-effects model.

The weight function, denoted as $w(Y_i, \sigma_i^2, Z_i)$ above, can be considered a function of *p*-value (p_i) and group membership (Z_i). Consider a weight function with *J* intervals in which, within each interval, the likelihood of survival is constant. Denote the left and right endpoints of the *j*-th interval (*a*) as a_{j-1} and a_j, respectively. The interval a_1, or the first interval, has a lower bound of $a_0 = 0$; the last interval a_k has an upper bound of $a_k = 1$. These bounds exist because the intervals describe a distribution of *p*-values, and *p*-values cannot be smaller than zero or larger than one. If the one-tailed *p*-value of a given study *i* falls within the *j*-th such interval, its weight is denoted ω_j. Then:

$$w(p_i, Z_i) = \begin{cases} \omega_1 & if\ 0 < p_i \leq a_1; \\ \omega_j & if\ a_{j-1} < p_i \leq a_j\ and\ Z_i = 0\ or\ a_j \leq 0.05 \\ \omega_j\lambda & if\ a_{j-1} < p_i \leq a_j\ and\ Z_i = 1\ and\ a_j > 0.05 \\ \omega_k & if\ a_{k-1} < p_i \leq 1\ and\ Z_i = 0\ or\ a_k \leq 0.05 \\ \omega_k\lambda & if\ a_{k-1} < p_i \leq 1\ and\ Z_i = 1\ and\ a_k > 0.05 \end{cases}$$

Because the number of studies present in each interval prior to censorship is unknown, the weights are relative rather than absolute; to compensate for this indeterminacy in a maximum-likelihood context, the weight for the first *p*-value interval (ω_1) is constrained to 1.0. Multiplying the weights for given intervals by λ allows the weights for those intervals to vary proportionally while estimating only one additional parameter.

Readers may wonder why λ applies only to the weights for the nonsignificant *p*-value intervals. Theoretically, of course, it is possible to apply λ to all the estimated weights, regardless of statistical significance. Because λ is a multiplicative constant, however, doing so would mean that the shift in the probability of survival is constant across both significant and nonsignificant *p*-values, which is highly unlikely.

Consider a hypothetical example. In this scenario, the meta-analyst specifies *p*-value cutpoints at 0.01, 0.05, 0.10, 0.20, 0.50, and 1.00, resulting in six *p*-value intervals: $0 < p < 0.01$, $0.01 < p < 0.05$, $0.05 < p < 0.10$, $0.10 < p < 0.20$, $0.20 < p < 0.50$, and $0.50 < p < 1.00$. Assume that, for one group of effect sizes (coded as 1), the parameter values of the weights are, respectively, 1.00, 1.00, 0.90, 0.80, 0.70, and 0.50. This represents a situation where studies with *p*-values below 0.05 always survive selection and where the likelihood of survival decreases as studies' *p*-values increase, such that studies with *p*-values above 0.50 are only half as likely to survive as those with *p*-values below 0.05. Now assume that a second group of effect sizes (coded as 0) has parameter values of 1.00, 1.00, 1.00, 1.00, 1.00, and 1.00. In other words, this second group of effect sizes represents a case in which studies always survive selection regardless of their *p*-value. If we allow λ to be a multiplicative constant of the significant *p*-value intervals, λ would be estimated at 1.00, indicating that the pattern of survival is the same across groups – which it is for the significant intervals (the first two). This specification of the model would not be able to reflect the differing patterns of selection. If λ was applied to the nonsignificant intervals, however, it would be estimated at approximately 0.73 – the

average of the four nonsignificant weights. This informs the meta-analyst that, for one group, the chances of surviving selection are lower for studies with nonsignificant *p*-values. Although it is not a perfect representation – 0.73, for instance, underestimates the weight of 0.90 and overestimates the weight of 0.50 – applying λ to the nonsignificant *p*-value intervals yields a much more accurate representation.

Recall that λ is applied to the group of studies that are coded 1 on the relevant moderator variable. Either group can be coded 1. However, it sometimes makes more sense to code the group that is likely to be less biased as zero, which results in a smaller estimate of λ. Coding the more biased group as 1 is possible but may sometimes result in cases where λ is estimated at a large and unwieldy value. This coding decision does not fundamentally change the model.

The general issue of publication bias exists, in part, due to the reliance of the research community on null hypothesis significance testing (NHST). It is, therefore, ironic to base our assessment of publication bias on the results of a statistical test. However, researchers continually attempt to do so. For those who favor NHST, it is possible to formally compare the Vevea and Hedges (1995) model results to the lambda model results using a likelihood-ratio test. The meta-analyst can obtain the likelihood value for the original Vevea and Hedges model and compare it to the likelihood value of the lambda model. This test will have one degree of freedom, representing the one parameter that is fixed – λ. If the test is significant, the meta-analyst has a piece of evidence indicating that a model allowing selection to vary across groups is "better" than a model where selection is fixed.

Although I demonstrate the use of likelihood-ratio tests in the substantive example, I do not explore the performance of them in the simulations, simply because their use is not the focus of the model. The most important aspects of the model results are its recovery of the population mean estimate and its estimate of the difference in bias patterns across groups (λ), and both estimates are explored via simulation.

The lambda model allows selection-bias patterns to vary across groups of effect sizes while including as few additional parameters as possible. It also allows the variance component (τ^2) and the overall mean, or conditional means (β), to remain constant across groups. The two subsequent sections, respectively, demonstrate its use in a substantive example and explore its performance through simulation.

4.1 Example

For the *R* code used to conduct these analyses, see Appendix C. Note that installing the *R* package *weightr* is required (Coburn & Vevea, 2015).

Recall that this example involves Daryl Bem's work on precognition. Studies published before attention was drawn to the "replication crisis," or before 2011, display asymmetry, indicating that a relationship is present between study size and effect size. Studies published after 2011, during the "replication crisis," display a distinct *lack* of asymmetry. See Figure 1 and Figure 2 for the funnel plots of these groups. This is an ideal case for the lambda model. We have reason to believe that year of publication may impact selection bias, but it would make absolutely no sense for the conditional mean to differ across year of publication – unless, of course, some event occurred in 2011 that

impacted the world's precognitive powers. Life is not an episode of The X-Files (Carter, 1993-2002) so we proceed to the lambda model.

I coded the later studies as 0, and the earlier studies as 1. I set *p*-value cutpoints at $p = 0.025, 0.05, 0.10, 0.50, 0.90$, and 1.00, resulting in a total of six *p*-value intervals (or five estimated weights). I included $p = 0.025$ because a one-tailed *p*-value of 0.025 represents one tail of a two-tailed alpha level of 0.05; 0.05 and 0.10 are also psychologically significant *p*-values, and 0.90 is the opposite tail corresponding to 0.10. $p = 0.50$ represents the point at which most effect size metrics switch from positive to negative. I would have included a few other cutpoints – perhaps 0.950 and 0.975 – but there were only one or two observed effect sizes with *p*-values in that range, which is too little information for the model to converge.

The unadjusted parameter estimates are featured in Table 2, and the adjusted parameter estimates in Table 3. There is no variance-component estimate for the adjusted model, because the variance component is reduced so much that it is essentially zero. This results in a border condition, where the estimate is too close to its lower bound, meaning that the model cannot estimate the variance component – therefore, the adjusted parameter estimates are obtained from a fixed-effect model. Random-effects estimates are equal to fixed-effect estimates if the variance component is zero, so it is acceptable to estimate a fixed-effect model here.

Table 2. Unadjusted random-effects meta-analytic parameter estimates, Bem data.

Parameter	**Estimate**	**Standard Error**	**Z-statistic**	***p*-value**	**CI Lower Bound**	**CI Upper Bound**
Intercept (β_0)	0.07137	0.01418	5.033	.0000005	0.04358	0.09917
Variance Component (τ^2)	0.00370	0.00230				

Table 3. Lambda model parameter estimates, Bem data.

Parameter	Estimate	Standard Error	Z-statistic	*p*-value	CI Lower Bound	CI Upper Bound
Intercept (β_0)	0.02283	0.01513	1.509	0.131307	-0.006823	0.05248
0.025 < *p* < 0.05 (ω_2)	0.86589	0.39382	2.199	0.027901	0.094013	1.63777
0.05 < *p* < 0.10 (ω_3)	0.47020	0.26764	1.757	0.078947	-0.054367	0.99477
0.10 < *p* < 0.50 (ω_4)	0.24681	0.13440	1.836	0.066299	-0.016607	0.51024
0.50 < *p* < 0.90 (ω_5)	0.28793	0.18818	1.530	0.125986	-0.080886	0.65675
0.90 < *p* < 1.00 (ω_6)	0.26683	0.23848	1.119	0.263193	-0.200582	0.73424
Lambda (λ)	0.54729	0.29048	1.884	0.059554	-0.022042	1.11663

The adjusted intercept estimate is approximately 0.02, compared to the unadjusted estimate of 0.07; this is a reduction of about 68%. Although precognitive studies tend to yield small effect sizes, a mean of 0.02 is objectively negligible. In fact, the adjusted mean is so small that it is not statistically significant ($p > .05$), and a difference of 0.02 standard deviations on average in response time is not likely to be of practical l significance either.

To interpret the weights and λ, recall that the later studies were coded 0, so the nonsignificant weights for the earlier studies must be multiplied by λ. The later studies do display a typical selection-bias pattern; studies with *p*-values between 0.025 and 0.05 are about 87% as likely to survive as those with *p*-values less than 0.025, those with *p*-values between 0.05 and 0.10 are about 47% as likely, and so on. It is interesting to learn that, despite the low degree of asymmetry present in the funnel plot, publication bias may still be present among the studies published after 2011. The question, however, remains – does the *pattern* of bias differ across groups?

For the earlier studies, the weights for the first and second *p*-value intervals (1.00 and 0.87, respectively) remain the same, because both intervals are significant ($p < .05$). The other weights are multiplied by λ, which is estimated at approximately 0.55. For example, the later studies have a weight of 0.47 for *p*-values between 0.05 and 0.10. Earlier studies have a weight of 0.26 (0.47 x 0.55) for that same interval. The chance of survival for studies published before 2011 is half that for studies published after 2011. That is a drastic difference. Although publication bias appears to affect both earlier and

later studies, the pattern is much more severe in studies published before 2011 – prior to the media storm sparked by Bem's first studies (2011).

We can also conduct the likelihood-ratio test described above. For the Bem et al. (2016) data, the test is nonsignificant, $\chi^2(1) = 0.6432$ ($p = 0.42$). This indicates that the lambda model and the original Vevea and Hedges (1995) model fit the data equally well.

Although the likelihood-ratio test is nonsignificant, the test alone is not conclusive proof that the pattern of bias does not differ. The λ estimate is substantively less than one and the funnel plots are drastically different. Including λ results in slightly better model fit, just not significantly better. The results of this statistical test can also vary depending on the intervals estimated. The lambda model, along with its Vevea and Hedges (1995) counterpart, is fundamentally a sensitivity analysis; hence, researchers should be reluctant to come to a general conclusion based on one significance test.

If researchers are in doubt about the necessity of the lambda model, I invite them to note the models' nested structure. The lambda model will collapse to the Vevea and Hedges (1995) model if there is truly no difference in the selection pattern; λ will equal one. Therefore, if the bias pattern does not differ, using the lambda model should not change anything; if it *does* differ, even slightly, using the lambda model will provide a better estimate.

Thus far I have used terms like "should," "likely," and so on. I wish to emphasize that we have been working with one substantive dataset. To draw conclusions about the model's future performance and overall abilities, we should evaluate its performance empirically. The next section of this chapter describes the results of a simulation to this end.

4.2 Simulation Results

For each cell, I conducted approximately 10,000 replications. I say "approximately" because in some cells (especially when the sample size, *k*, is small and there is no heterogeneity) the models occasionally have trouble converging, resulting in the completion of fewer than 10,000 replications. When this happens, however, it is only due to the presence of a border condition (a variance component near zero), and it rarely occurs. For the vast majority of cells, 10,000 replications were conducted.

The next sections present assessments of simulation convergence, followed by the simulation results for the lambda model, broken down by parameter. I focus on the adjusted estimate of the mean and the estimate of λ, which are the primary parameters of interest. I present plots of the parameter estimates and their root mean squared error (RMSE) and discuss the model's performance. I also compare its estimate of the mean to that from a completely unadjusted meta-analytic model and to that from the original Vevea and Hedges (1995) model.

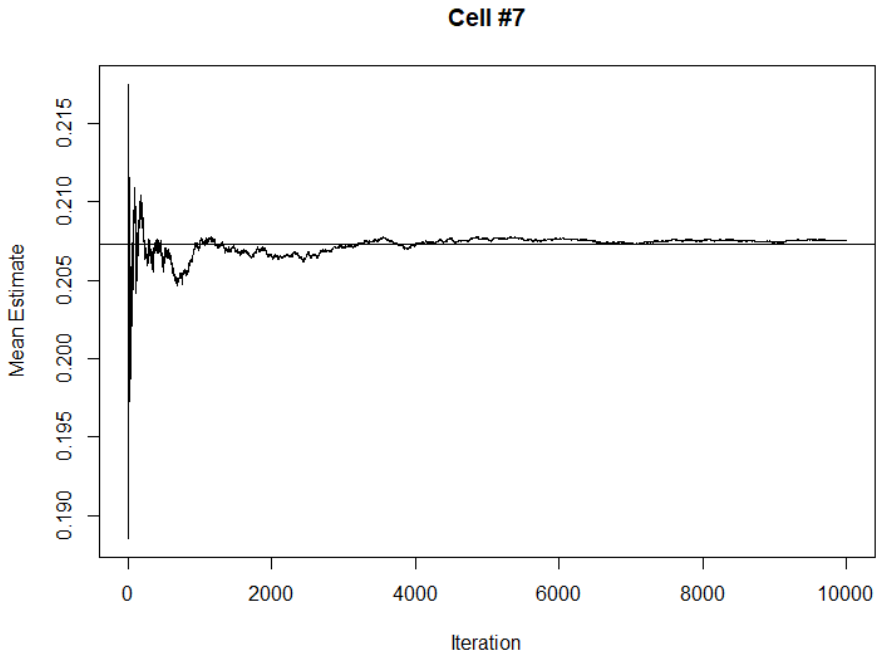
4.2.1 Convergence

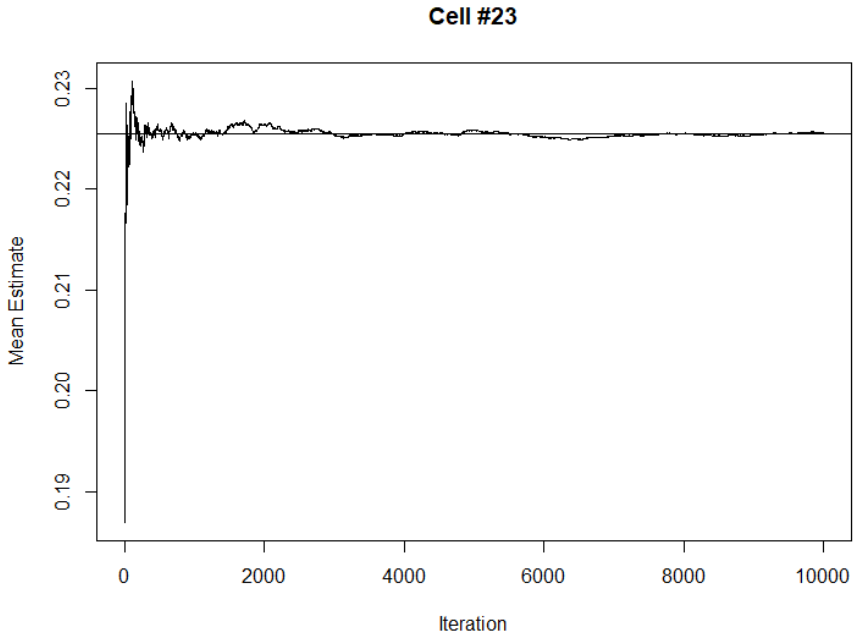
In this section, note that I am not referring to whether the numerical optimization methods of model estimation converged on a parameter estimate. Those issues of failed optimization are discussed above – they occur rarely, and only in the presence of a border

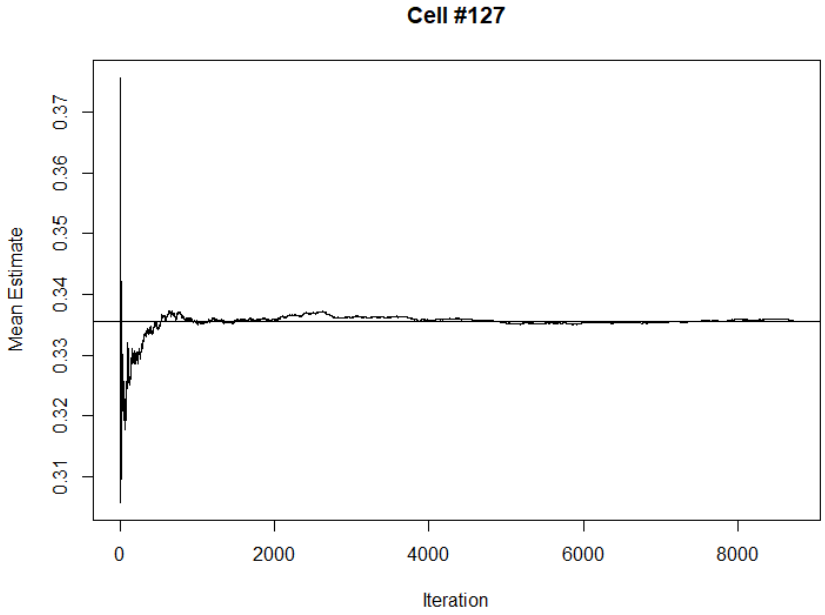
condition. Here, I am interested in determining whether the *simulation results* have converged. If the cumulative mean of a given parameter estimate is fluctuating violently across cell replications, that cumulative mean is meaningless and cannot be treated as a representative estimate in assessments of bias and RMSE.

To assess convergence of the cumulative mean, I plot an estimate of the cumulative mean against the corresponding number of cell replications. When only a few replications have been completed (i.e., toward the left of the plot), the mean will fluctuate. However, by close to 10,000 replications, the plot will ideally be a straight line, indicating that the cumulative mean has stabilized (Coburn & Vevea, in prep). The presence of such a straight line does not absolutely guarantee convergence of the cumulative mean, but it is not evidence of non-convergence, and therefore we can assume that the mean is a viable estimate.

Plots of the cumulative mean estimate for four cells are presented in Figure 12.




Cell #23
Mean Estimate
0.19
0.20
0.21
0.22
0.23
0
2000
4000
6000
8000
10000
Iteration


Cell #127
Mean Estimate
0.31
0.32
0.33
0.34
0.35
0.36
0.37
0
2000
4000
6000
8000
Iteration

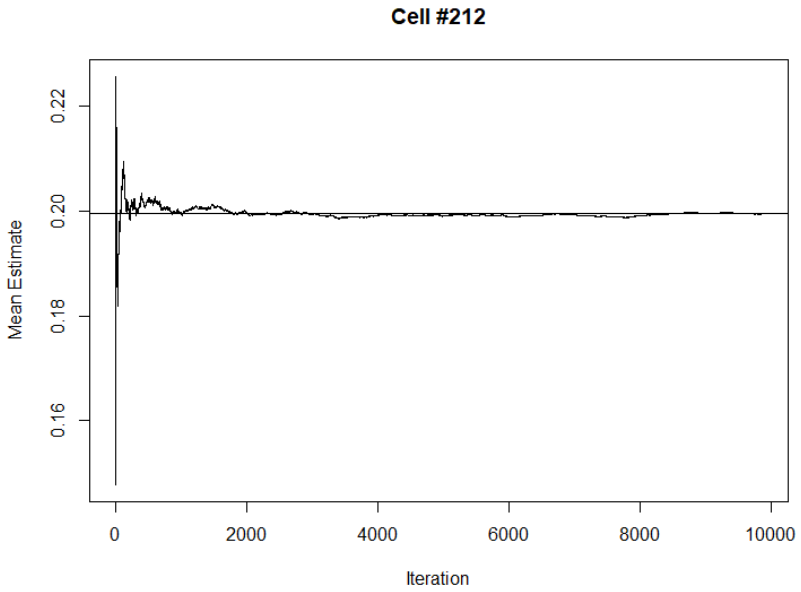


Figure 12. Plots of the cumulative mean for the lambda model.

Plots from the other cells are not presented here for the sake of brevity. However, these plots are a representative example of the results across cells. There is no strong sign of non-convergence; therefore, we can proceed.

4.2.2 Mean Estimate

Across all cells, the parameter value of the overall mean prior to any selection mechanism is 0.20. The plots presented in this section compare the cumulative mean estimate, across 10,000 replications per cell, of each of three models – an unadjusted random-effects meta-analysis, the Vevea and Hedges (1995) model, and the lambda model. The population mean (pre-selection) is represented by a single horizontal line at 0.20. The three models are represented by different line types; the lambda-model estimates are a solid line, the unadjusted estimates a line with two dashes, and the Vevea and Hedges (1995) estimates a dotted line. Points represent cell means.

First, I examine estimates of the mean in cells where the lambda model is likely to perform best – that is, in cells where the selection mechanism exactly matches the model. Figure 13 displays the results for cells where I^2 is 0% (that is, there is no between-studies heterogeneity) and in which selection was generated according to the model. The four panels represent bias patterns – "None," "None vs. Weak," and so on. Levels of *k* are presented along the *x*-axis.

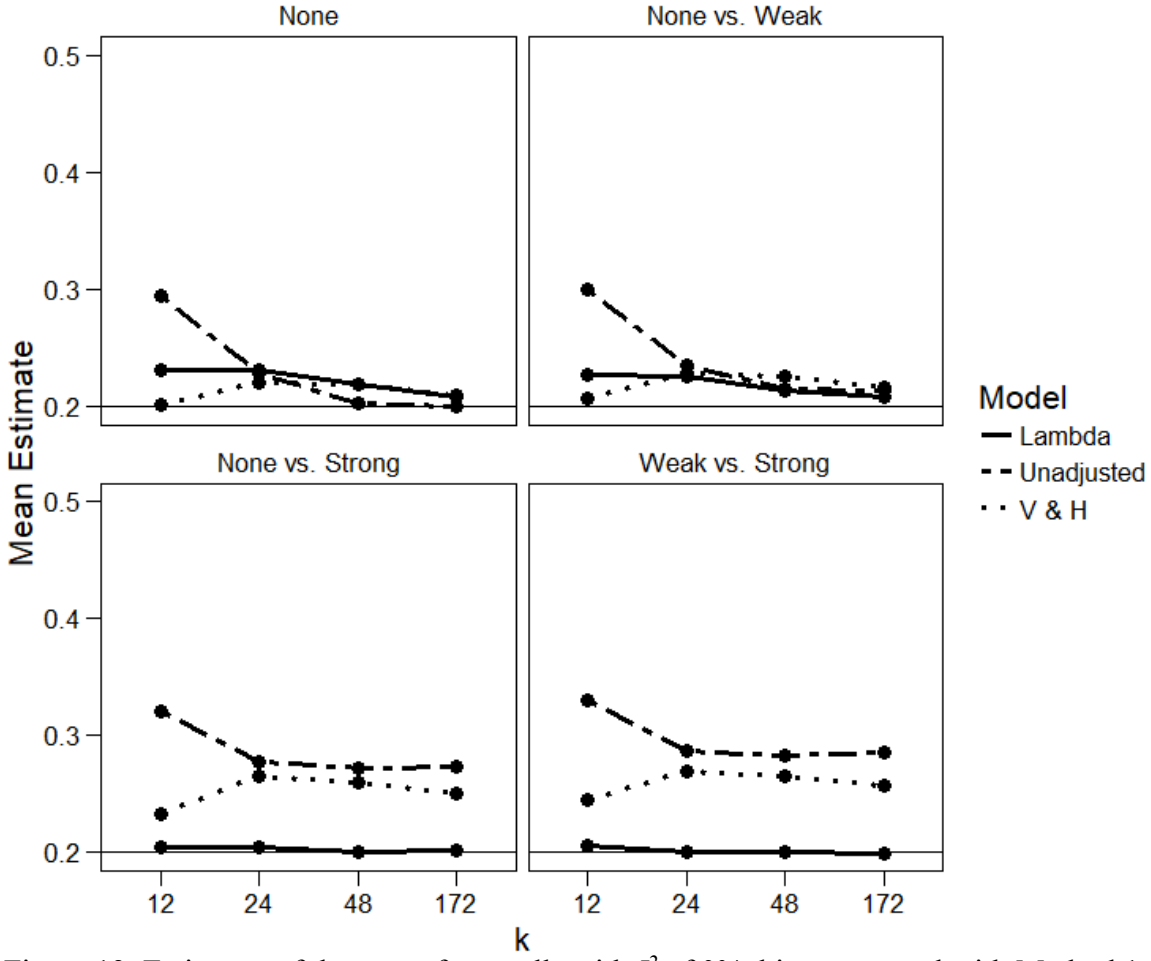


Figure 13. Estimates of the mean from cells with I^2 of 0%, bias generated with Method 1.

When I^2 is 0% and no bias is present in either group (the top left panel of Figure 13), the unadjusted model yields the most inflated estimate when *k* is 12, but by the time *k* reaches 172 the unadjusted model actually produces the least inflated estimate of the population mean. However, all three estimates are very close, which is a positive outcome overall – if no bias is present, the lambda model is not quite the best estimate, but it is still very close.

When weak bias is present in one group and I^2 is 0% (the top right panel of Figure 13), the pattern is similar to that in the top left panel, but the lambda model no longer inflates the mean as drastically. When strong bias is present in one group, though (both panels in the bottom row of Figure 13), the lambda model produces the least inflated estimate of the mean by far. Its estimate is virtually unbiased, even in cases where *k* is as

small as 12. It is worth noting that the Vevea and Hedges (1995) model yields a more accurate estimate of the mean than the unadjusted model when strong bias is present in one group, although this estimate is nowhere near as accurate as the lambda model estimate (which accommodates the difference across groups).

For cells in which I^2 is 25% (a variance component of 0.03) and in which bias was generated according to the lambda model (Method 1), see Figure 14.

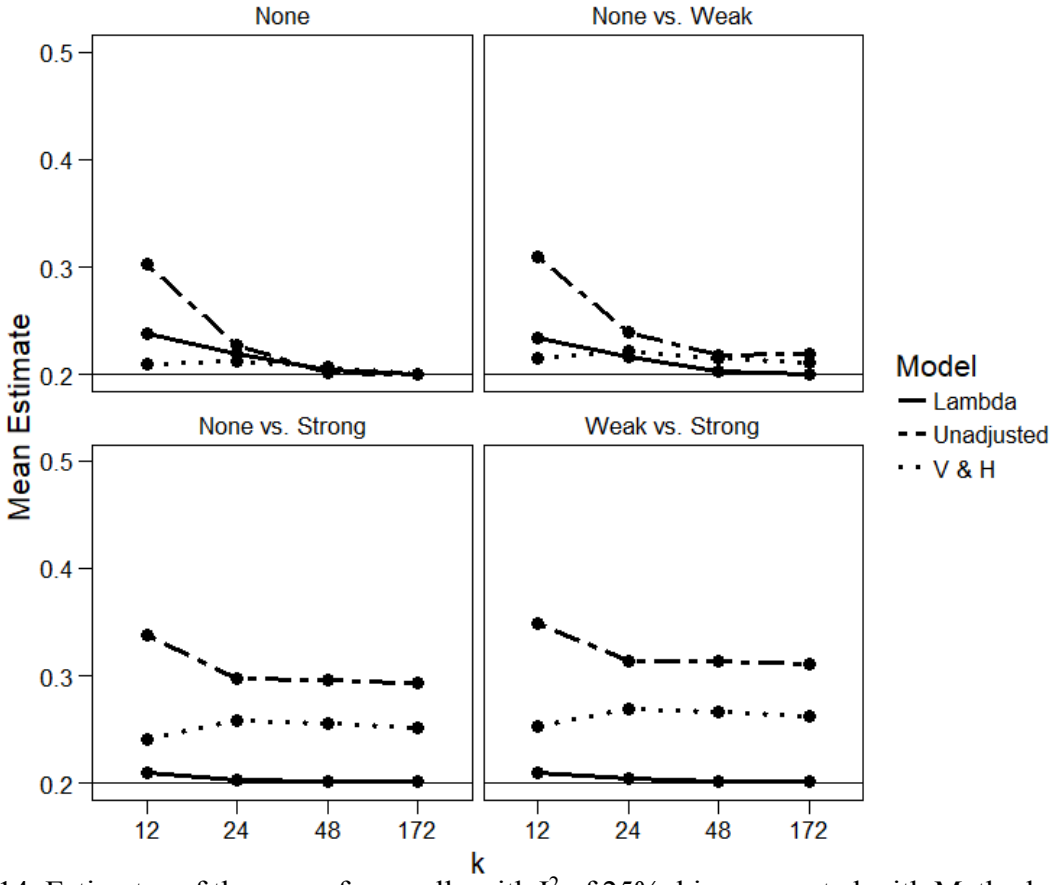


Figure 14. Estimates of the mean from cells with I^2 of 25%, bias generated with Method 1.

Now that some between-studies heterogeneity is present, the performance of the lambda model has improved in the top left panel, where no bias is present in either group. Its performance matches that of the unadjusted model by the time *k* is 172, yielding a virtually unbiased estimate of the mean. When weak bias is present in one group (the top right panel), the lambda model produces a more accurate estimate of the mean than either of the other two models by the time *k* reaches 24. Finally, for strong bias (the bottom two panels), the lambda model again performs flawlessly, with only very slight inflation when *k* is 12. One noticeable difference between Figure 13 and Figure 14, however, is that the Vevea and Hedges (1995) model also does a better job with I^2 of 25% versus 0%; note that its estimate is less inflated in the bottom panels of Figure 14 than in Figure 13. The model is fundamentally incapable of accounting for the differing bias pattern across groups, but it makes a valiant effort regardless.

Figure 15 shows cells in which I^2 is 50% (a variance component of 0.03) and in which bias was generated according to the lambda model (Method 1).

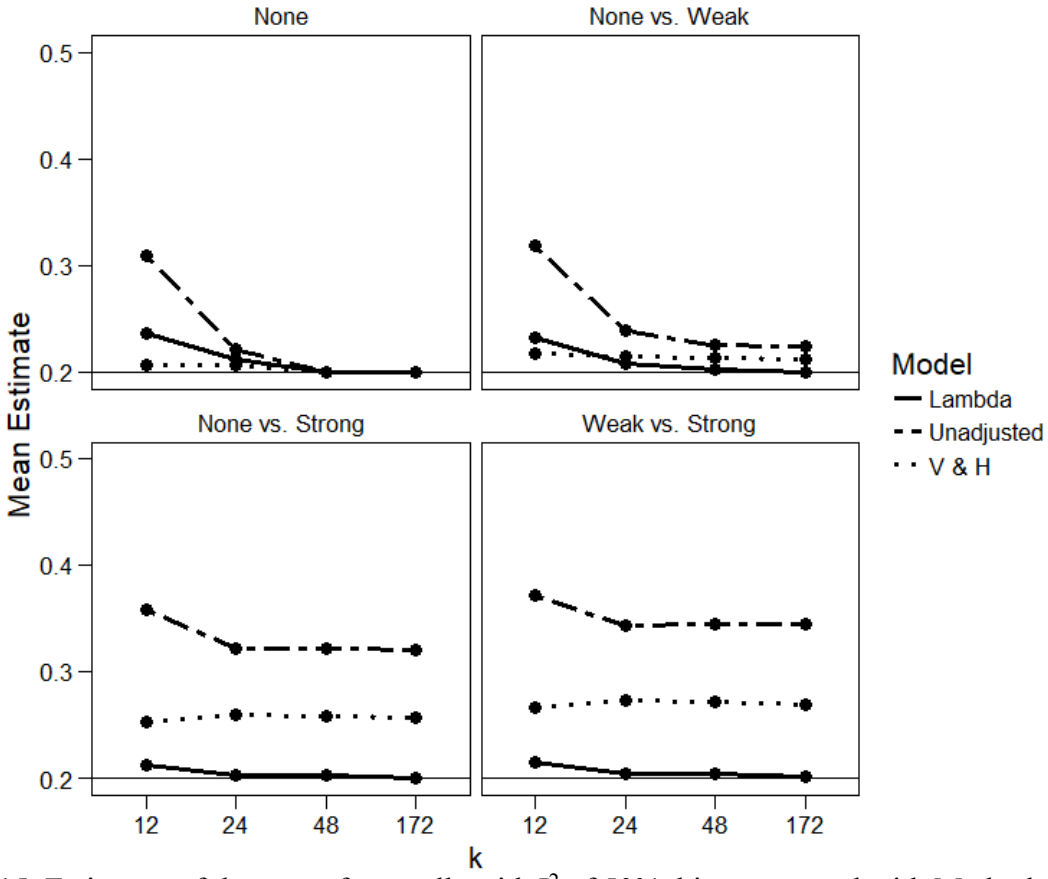


Figure 15. Estimates of the mean from cells with I^2 of 50%, bias generated with Method 1.

The pattern in this figure is the same as in the previous two. In the top left panel (no bias in either group), the lambda model does just as well as the other two models by the time *k* is 48. When weak bias is present in one group, the lambda model yields a better estimate of the mean than even the Vevea and Hedges (1995) model (top right panel). When strong bias is present in one group (bottom two panels), the lambda model produces a virtually unbiased estimate of the population mean across all levels of *k* (with the slight exception of *k* of 12). The Vevea and Hedges (1995) model again generally does better than the unadjusted meta-analytic model, but the lambda model outperforms both.

Finally, for cells where I^2 is 75% (a variance component of 0.23) and bias is generated according to the lambda model, see Figure 16.

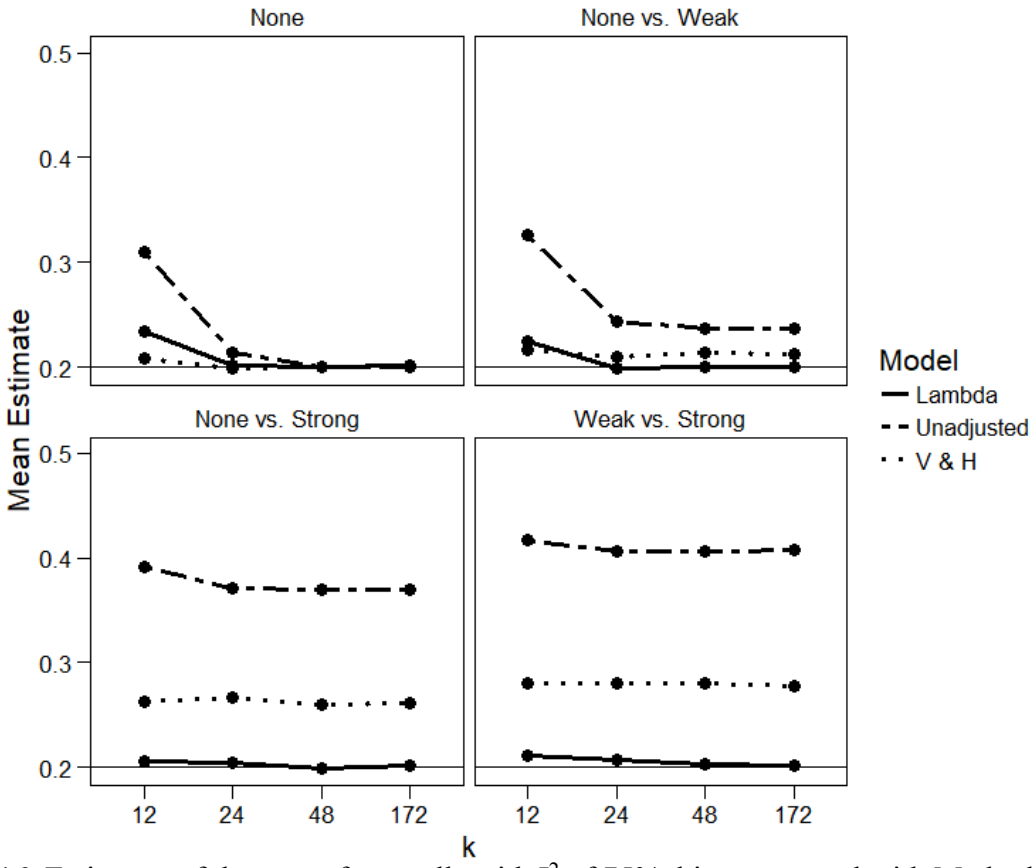


Figure 16. Estimates of the mean from cells with I^2 of 75%, bias generated with Method 1.

The pattern continues to hold. In the presence of heterogeneity – even at high levels, like an I^2 of 75% – the lambda model vastly outperforms both of its competitors. Of course, these figures have so far only addressed cases in which bias was generated according to the model's assumptions, and the results may vary across levels of bias generation, as the next figures will reveal.

Now that we have examined plots of the mean estimate across levels of bias pattern and I^2, we can examine variation in the mean estimate across levels of bias generation. Note that the facets now pertain to methods of bias generation. For these, and for most subsequent plots, to preserve space and eliminate unnecessary repetition, only cells with I^2 of 0% and I^2 of 75% are presented here. For interested readers, other plots are available in the Appendices (one Appendix per chapter).

Figure 17 shows cells where I^2 is 0% and the bias pattern is "None vs. Weak."

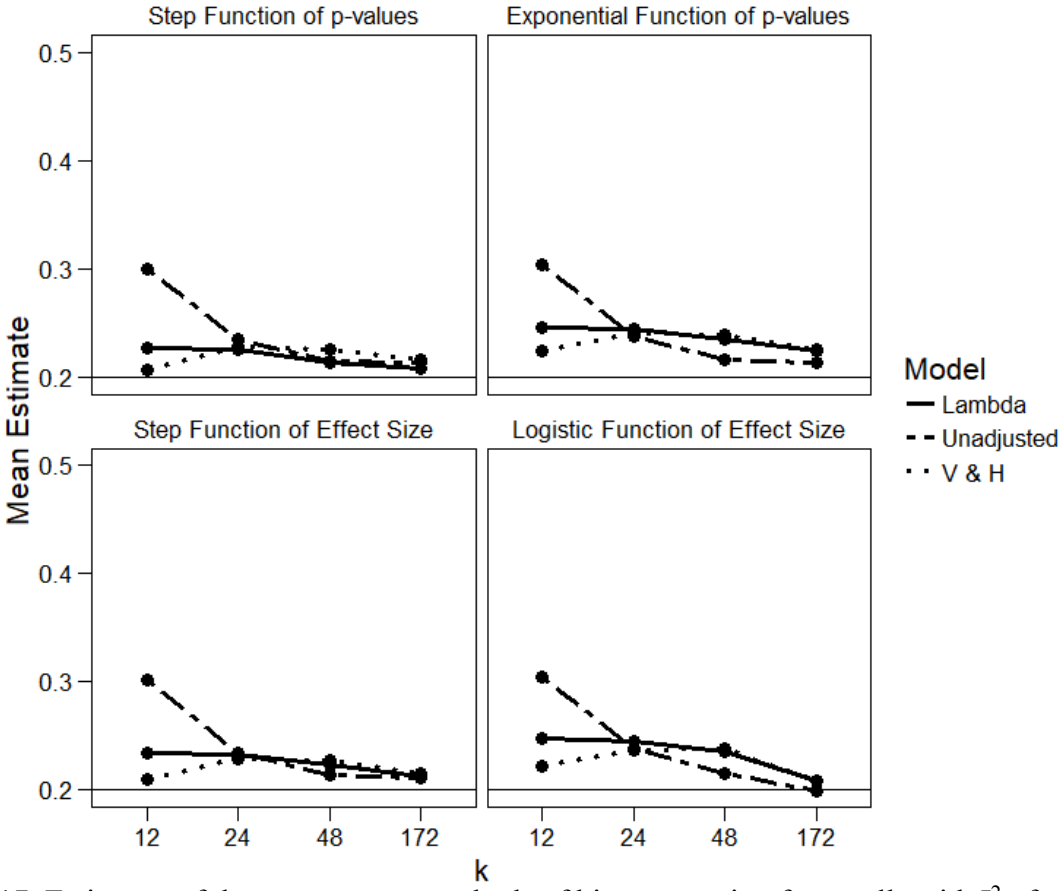


Figure 17. Estimates of the mean across methods of bias generation from cells with I^2 of 0% and "None vs. Weak" bias pattern.

Across all four panels here, apart from the top left panel when *k* is 48 and above, these are the cells in which the lambda model yields a more inflated estimate of the mean than the other two models. It is never drastically more inflated than the others, however; the biggest difference is in the two panels on the right, where bias is generated as an exponential function of *p*-values or a logistic function of effect sizes. This is perhaps to be expected, as weak bias in one group is the most difficult to detect and as the lambda model generally performs better than competing models in the presence of heterogeneity.

Figure 18 shows cells where I^2 is 75% and the bias pattern is “None vs. Weak.”

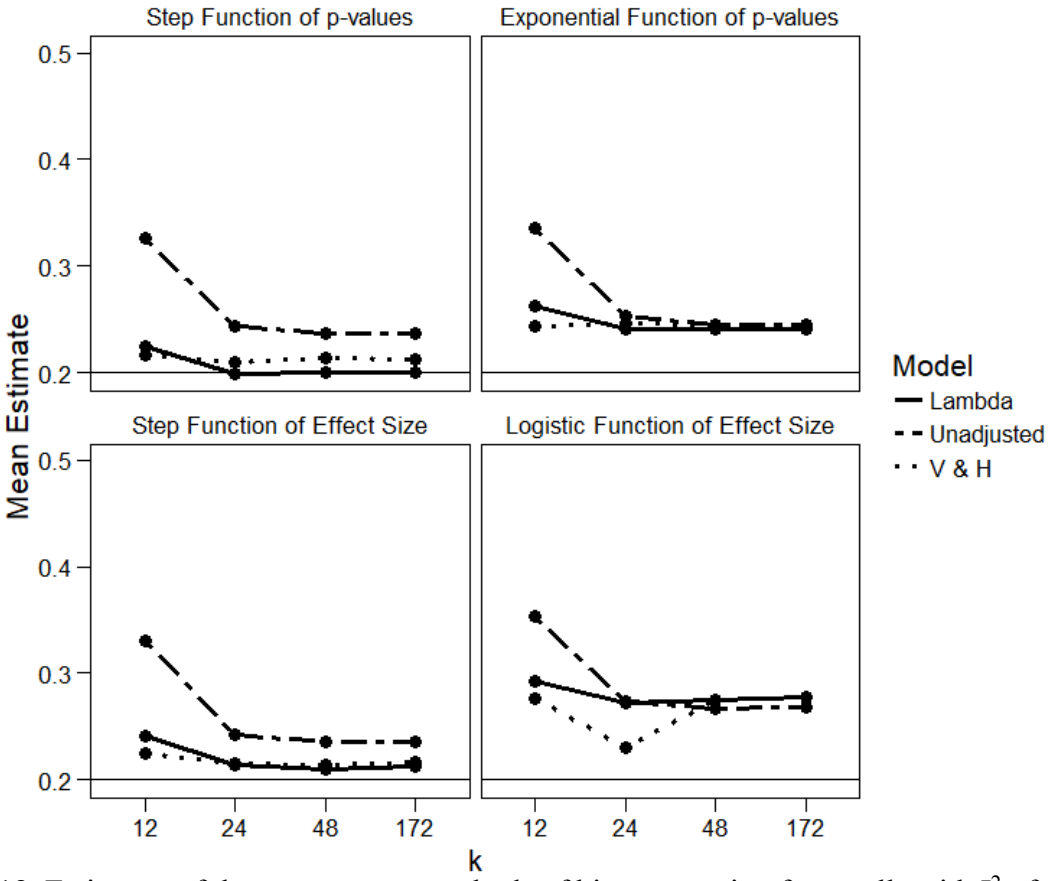


Figure 18. Estimates of the mean across methods of bias generation from cells with I^2 of 75% and "None vs. Weak" bias pattern.

In the presence of high heterogeneity, a slightly different pattern becomes clear. For the top left panel, the lambda model yields the least inflated estimate for all levels of *k* above 12; this is expected, however, because the top left panel is one where generation matches the model assumptions. In the bottom left panel, where bias is based on a step function of effect sizes, the lambda model and the Vevea and Hedges (1995) models perform about equally well. This is also somewhat to be expected, because weak bias in one group is a small enough difference that the Vevea and Hedges (1995) model is still mostly able to compensate. For the two panels on the right, all models yield a somewhat inflated estimate; the lambda and Vevea and Hedges (1995) models are usually about equal, although Vevea and Hedges (1995) is more accurate in a few places.

The “None vs. Weak” cells are likely the most difficult ones for the lambda model, as weak bias in one group has less of an effect and is less easy to detect than strong bias. The next plots explore situations where strong bias is present in one group. Figure 19 shows cells where I^2 is 0% and the bias pattern is “None vs. Strong.”

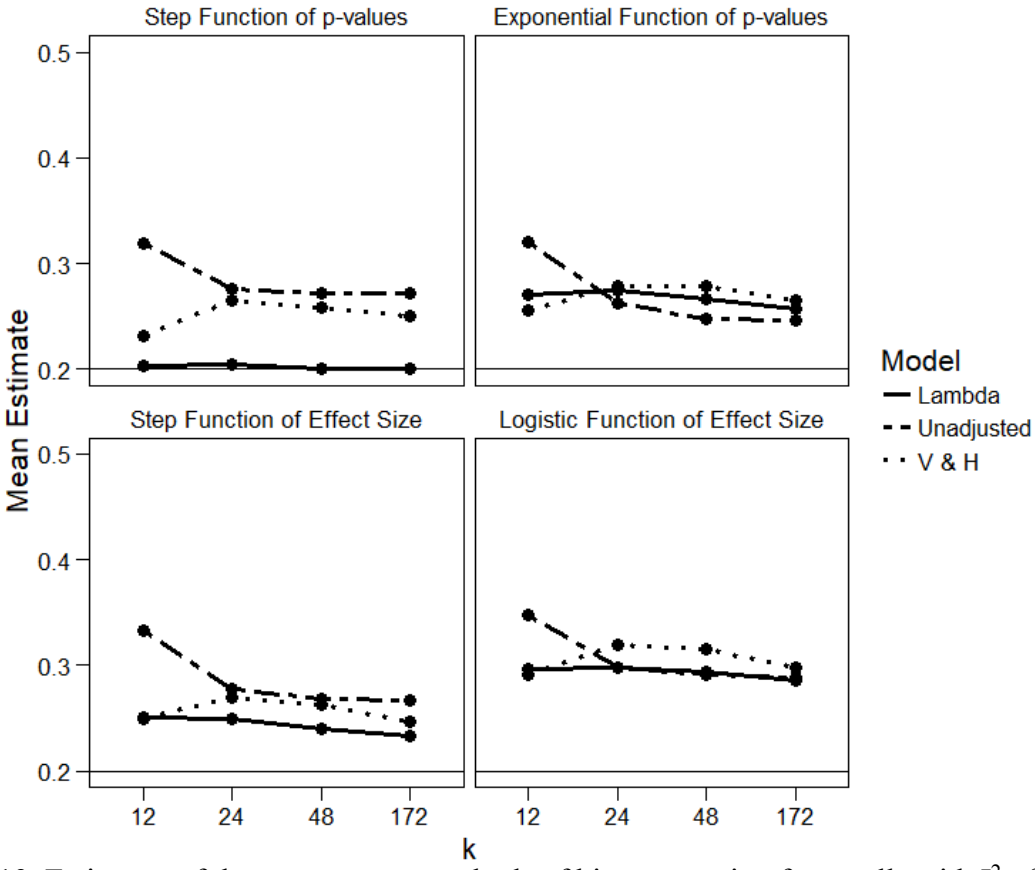


Figure 19. Estimates of the mean across methods of bias generation from cells with I^2 of 0% and "None vs. Strong" bias pattern.

For both panels on the left of Figure 19, the lambda model estimate is consistently less inflated than both the unadjusted model and the Vevea and Hedges (1995) model; bias generated as a step function, whether based on *p*-value or effect size, is closer to meeting the assumptions of the model than bias generated as an exponential or a logistic function. In the panel on the top right, the unadjusted model produces the most accurate mean estimate once *k* reaches 24. In the panel on the bottom right, where bias is generated as a logistic function, the lambda model and the unadjusted model both yield less inflated estimates (approximately equivalent) than the Vevea and Hedges (1995) model.

Finally, Figure 20 features a None vs. Strong bias pattern and I^2 of 75%:

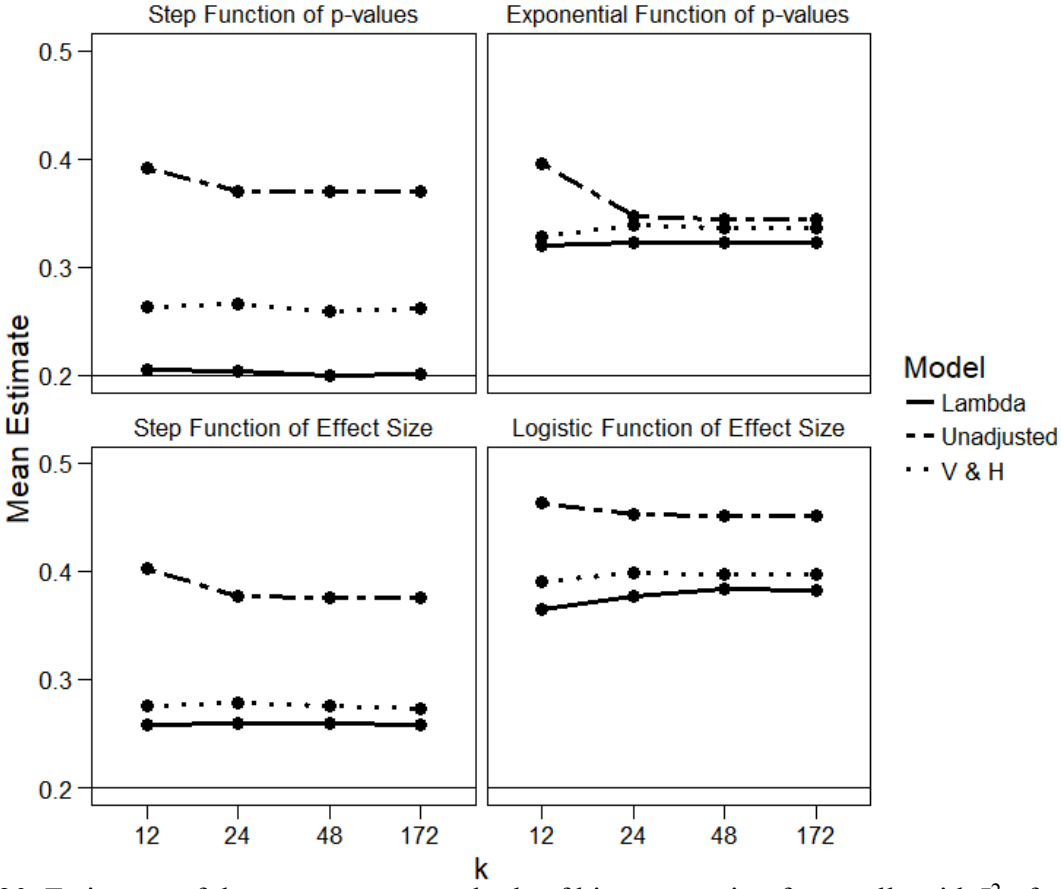


Figure 20. Estimates of the mean across methods of bias generation from cells with I^2 of 75% and "None vs. Strong" bias pattern.

Figure 20 demonstrates something very interesting – and encouraging. It demonstrates that, regardless of the method of bias generation and the sample size (even in cases where bias is an exponential function of *p*-value or a logistic function of effect size), if the data are heterogenous and strong bias is present in at least one group, the lambda model yields a more accurate estimate of the mean. Because many social science meta-analyses have large I^2 (Davis, Mengersen, Bennett, & Mazerolle, 2014) and most are likely subject to publication bias, these results are promising.

One bias pattern remains, “Weak vs. Strong.” This pattern seeks to explore the question of whether the model can adjust for cases in which bias is present in both groups, albeit to differing degrees. Figure 21 displays cells with I^2 of 0% and “Weak vs. Strong.”

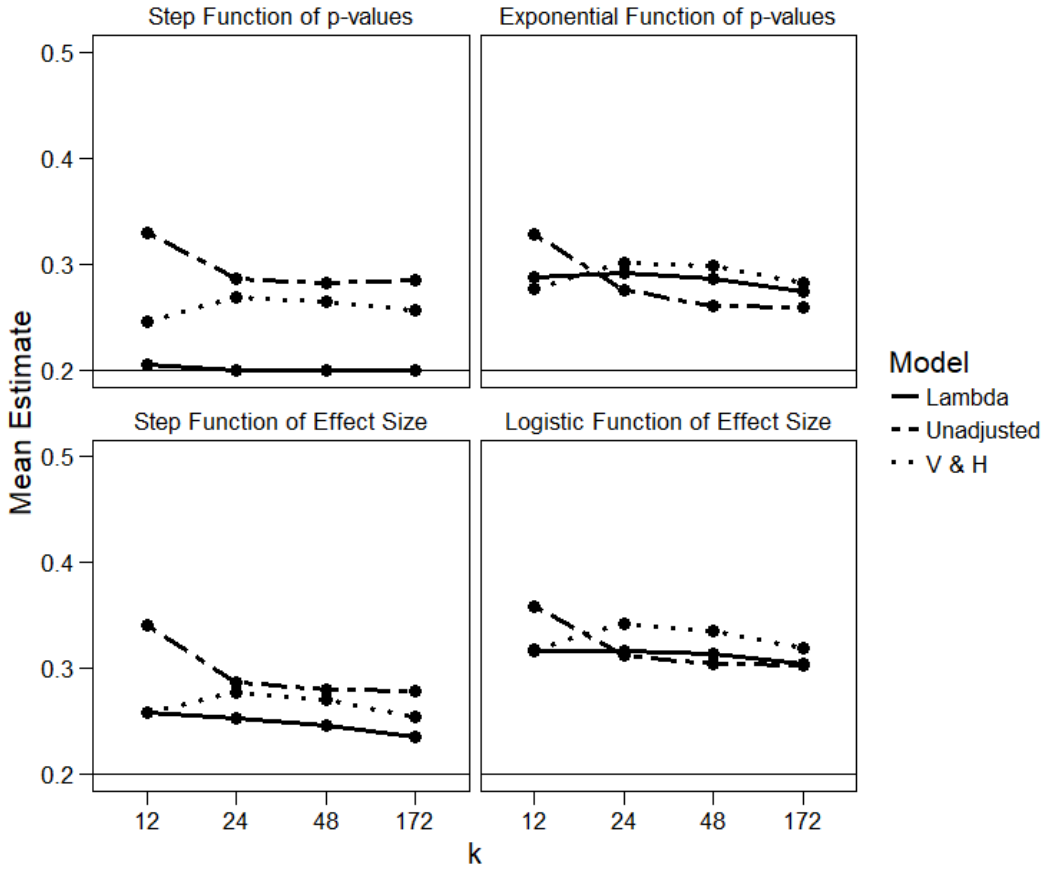


Figure 21. Estimates of the mean across methods of bias generation from cells with I^2 of 0% and "Weak vs. Strong" bias pattern.

For both panels on the left, when bias is generated as a step function, the lambda model yields a consistently more accurate estimate of the mean across levels of *k*.

Figure 22 displays cells with I^2 of 75% and “Weak vs. Strong.”

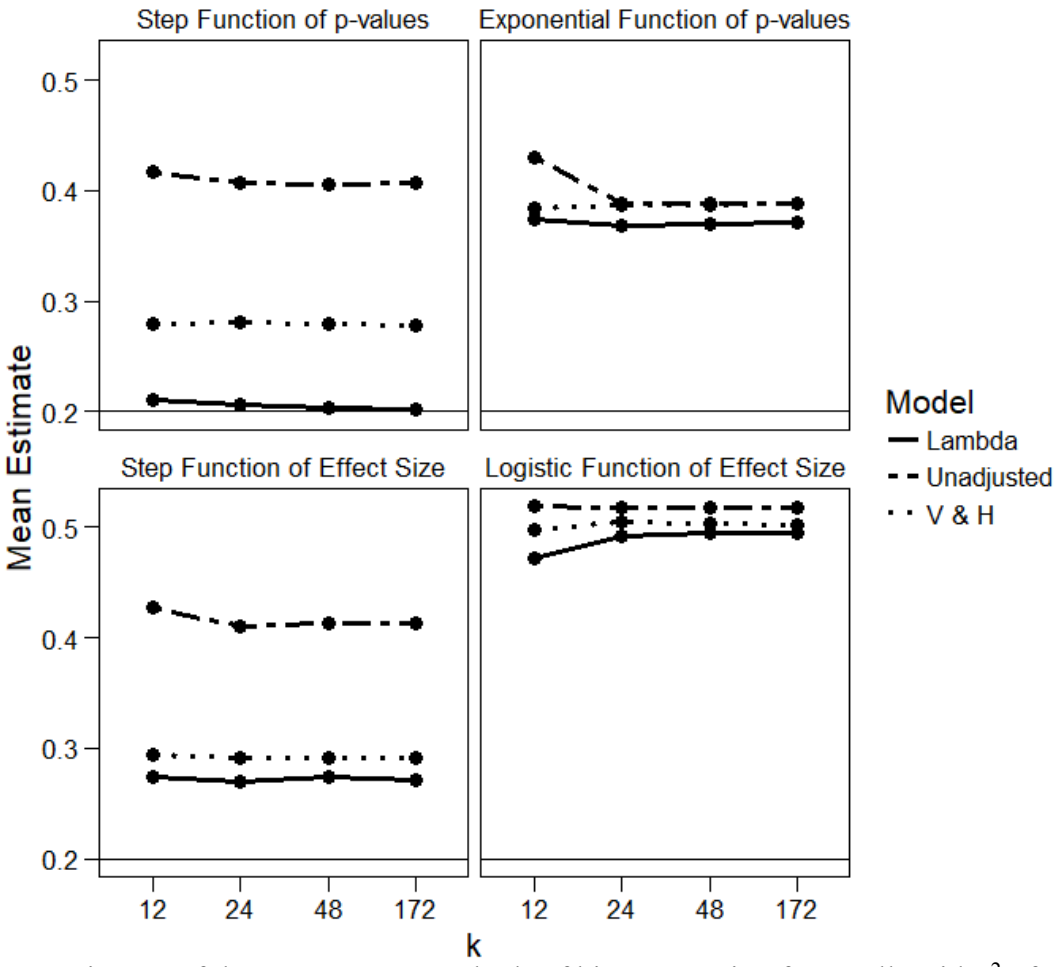


Figure 22. Estimates of the mean across methods of bias generation from cells with I^2 of 75% and "Weak vs. Strong" bias pattern.

Again, when there is high heterogeneity (I^2 of 75%), the lambda model really shines. Its performance is best when bias is generated according to the model (top left) and second best when bias is a step function of effect size (bottom left). The lambda model does still manage to yield a less inflated estimate than both other models even when bias is generated as an exponential function of *p*-value or a logistic function of effect size (right panels), but just barely. Compare the two right panels to the corresponding panels in Figure 20; notice that, when bias is not generated as a step function, the lambda model performs better if bias is present in one group, not both.

4.2.3 λ Estimate

The plots in this section are slightly more complicated. The population mean is the same across all cells and all conditions (0.20). λ, however, is an estimate of the difference in bias patterns across groups, as a multiplicative constant. This means that the true estimate of λ varies across levels of bias pattern.

For the "None vs. Weak" bias pattern, the population value of λ should be about 0.70 (one group has no bias, with a nonsignificant weight of 1.00, and the nonsignificant

weight for the second group is 0.70; 1.00 * 0.70 is 0.70). In the next figures, the observed "None vs. Weak" estimates are represented by a solid black line, and the predicted "None vs. Weak" estimates by a solid gray line. For the "None vs. Strong" bias pattern, the population value of λ should be about 0.30, for the same reasons; see Chapter 3 for details. A dashed black line represents the observed "None vs. Strong" estimates, and a dashed gray line the predicted estimates. The same pattern represents the "Weak vs. Strong" estimates, albeit with dotted lines. It is then possible to assess the accuracy of the λ estimate by the overlap of the black and gray lines. Figure 23 displays cells where I^2 is 0%.

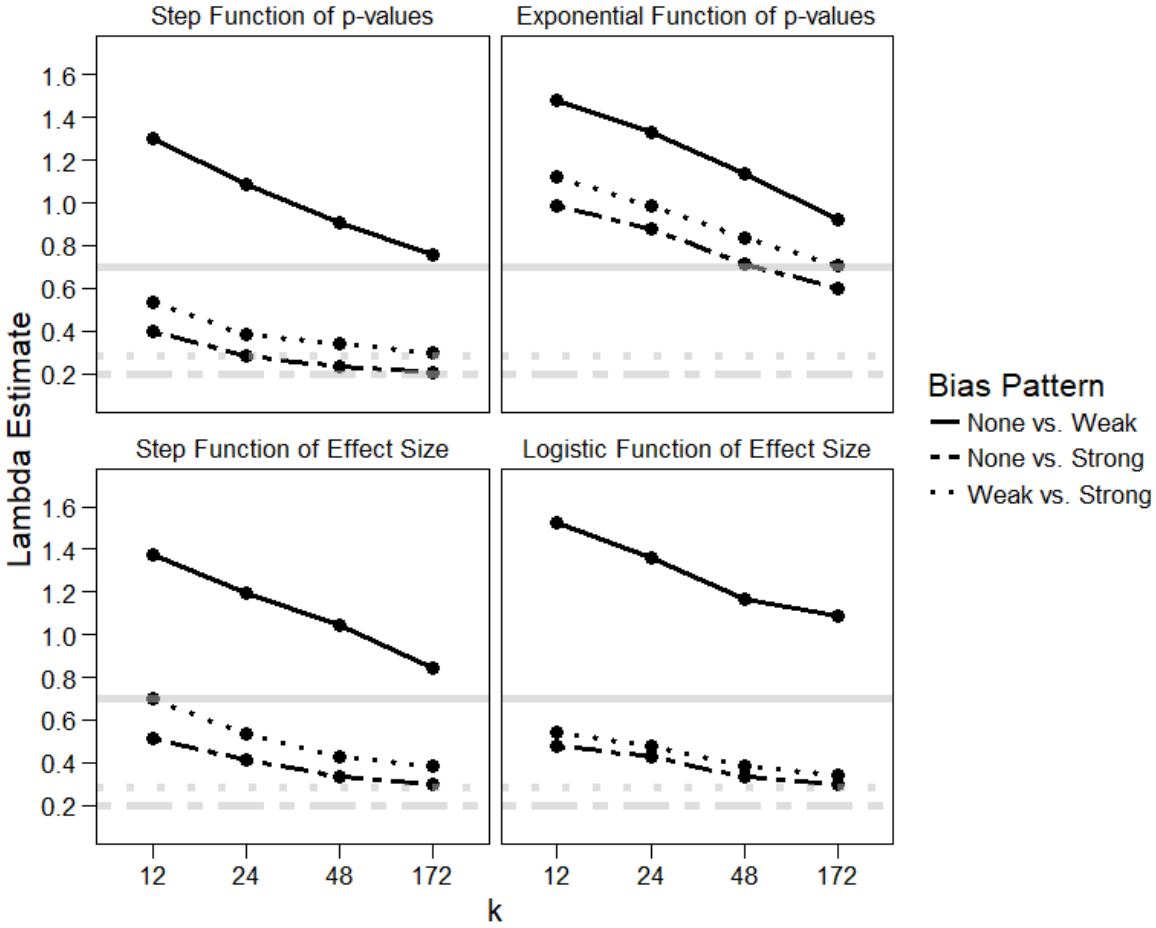


Figure 23. Estimates of λ across methods of bias generation and bias pattern from cells with I^2 of 0%.

In the top left panel, where bias is generated according to the model, the estimates of λ are almost completely uninflated by the time *k* is 172. In the bottom left panel, by *k* of 172, the λ estimate is still accurate, even though bias was generated as a step function of effect size. The model performs worse for the two panels on the right, as those generation mechanisms are very different from the model; however, the general pattern persists, which is somewhat surprising considering how different the bias generation is.

The same general pattern holds in Figure 24, which displays cells where I^2 is 75%.

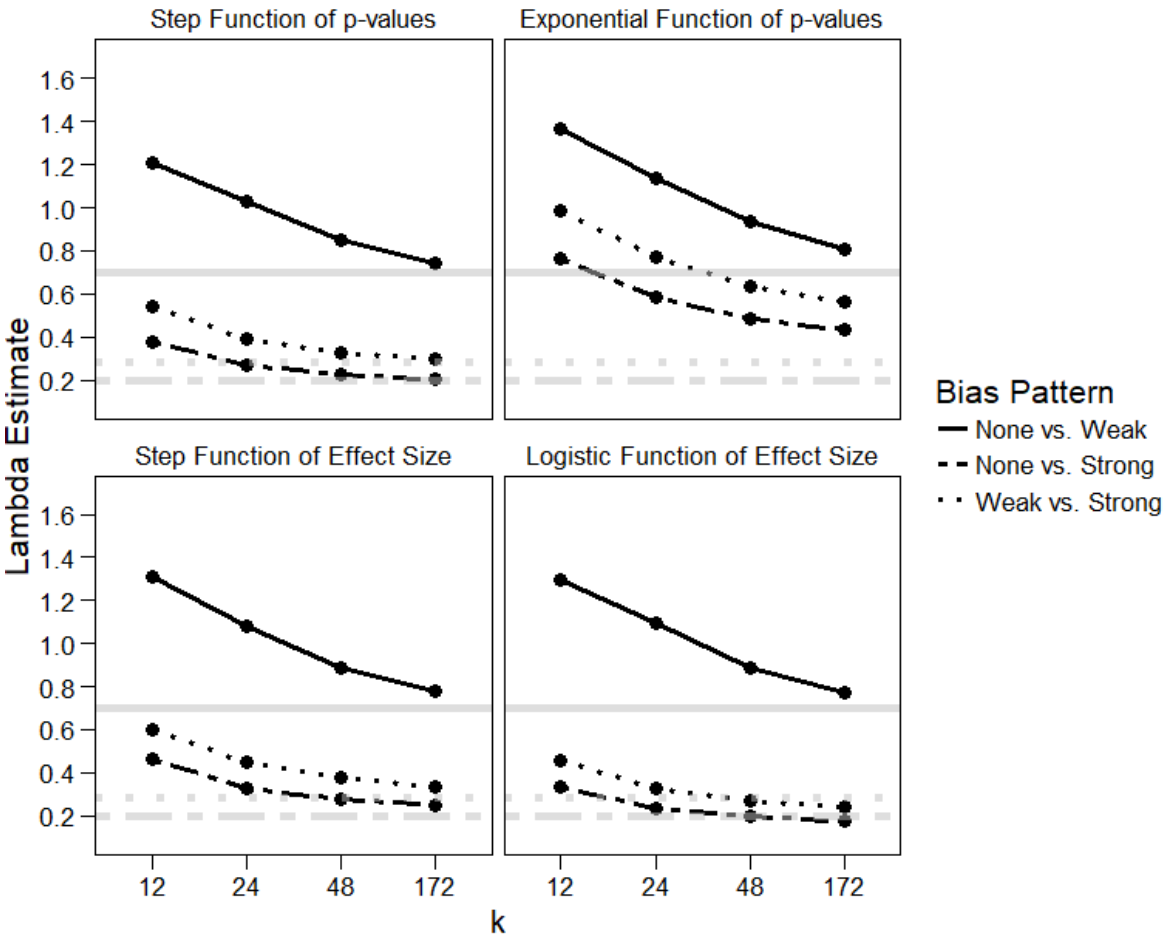


Figure 24. Estimates of λ across methods of bias generation and bias pattern from cells with I^2 of 75%.

Readers who compare the two right panels of Figure 23 and Figure 24 may notice that the lambda model continues to perform well in the presence of high heterogeneity; its estimate of λ is much more accurate when *k* is 172, despite the differing bias generation mechanisms.

4.3 Conclusions

I began this chapter by outlining the goals for an ideal model that can accommodate differing patterns of selection bias across levels of a moderator variable. I presented one such model, here dubbed the lambda model, that is a variation of the Vevea and Hedges (1995) weight-function model and explained why the lambda model meets my desired qualifications.

In the example section of this chapter, I used the Bem et al. (2016) empirical meta-analytic dataset to demonstrate that the lambda model can work with substantive data to illustrate differing patterns of selection. Then, in the simulation section, I discussed the results of an extensive simulation exploring the performance of the lambda model across levels of bias generation, bias patterns, sample size, and heterogeneity. This

simulation revealed that the lambda model generally yields a less inflated estimate of the population mean than both an unadjusted meta-analytic model and the Vevea and Hedges (1995) model across cells. The lambda model truly shines when between-studies heterogeneity increases – a heartening outcome, considering that so many social science meta-analyses feature a large degree of heterogeneity, and knowing that most other methods of bias assessment cannot account for heterogeneity at all (Vevea and Coburn, in prep).

The primary remaining problem with this model is its sample size requirement. Like the Vevea and Hedges (1995) model, the lambda model does require observed effect sizes in each specified *p*-value interval to converge, and meta-analysts may sometimes wish to specify more *p*-value cutpoints than are practical. The next chapter describes a lambda model that allows the meta-analyst to assess the effects of different fixed weight patterns on their data. Because this version does not estimate the weights, it circumvents the issue and will be useful for meta-analysts with smaller datasets.

Chapter 5: The Lambda Model as Sensitivity Analysis

Many meta-analyses in the published literature are relatively small, particularly in the medical field (Coburn, Vevea, & Orey, in prep), which makes estimation of the lambda model difficult. However, because the Vevea and Hedges (1995) model (and its lambda counterpart) has several desirable properties, Vevea and Woods (2005) presented a modification that both maintains some of the desirable properties of the original and is capable of use with small data sets. Their modification adopts the Vevea and Hedges (1995) model but circumvents the problem of directly estimating the weight function by imposing a set of fixed weights. These fixed weights are determined *a priori* and chosen to represent specific forms and severities of selection bias (Vevea & Woods, 2005). By applying a sequence of these models, with various sets of weights representing different types and severities of selection, meta-analysts can assess the impact of each selection pattern on their effect-size estimates and satisfy themselves that their results are robust to selection (Vevea & Woods, 2005). Because it fixes all the weight parameters, this model sacrifices the ability to produce valid standard errors and conduct likelihood-ratio tests. Nonetheless, it serves as a useful tool for assessing the impact of different selection patterns – and, most importantly, it does not require many effect sizes.

It is possible to implement a lambda version of the Vevea and Woods (2005) model as well. (Note: Although all publication bias models are sensitivity analyses by nature, I refer to this model as "the lambda model for sensitivity analysis" to differentiate it from the original version.) This version can take one of two forms: (1) all the weights and λ are fixed; (2) the weights are fixed and λ is freely estimated. The first version allows the meta-analyst to assess the change in the mean effect size across values of λ; additionally, λ is fixed, so the first version estimates only the mean(s) and variance component (the smallest number of parameters). However, the second version is more informative. The meta-analyst specifies a series of fixed weights, as before. Though only one additional parameter (λ) is estimated, the model can produce a "best guess" of the weight function in the second group, under the strict assumption that the differences in the nonsignificant weights across groups are constant. This allows the meta-analyst to assess the difference in bias patterns across groups while estimating only one parameter (in addition, of course, to a mean and variance component) and manipulating the baseline group at will.

Let us consider a theoretical example to clarify. Imagine a meta-analytic dataset with an unadjusted mean of $d = 0.20$. The effect sizes are divided into two groups according to some relevant study characteristic – "Group One" and "Group Two." The researcher specifies four *p*-value cutpoints, $p = 0.01$, 0.025, 0.10, and 0.50, resulting in five intervals, $p < 0.01$, $0.01 < p < 0.025$, $0.025 < p < 0.10$, $0.10 < p < 0.50$, and $0.50 < p < 1.00$. To demonstrate the Vevea and Woods (2005) model, the researcher also specifies a set of weights for these intervals, rather than estimating the weights. The weights specified are 1.00, 0.90, 0.60, 0.40, and 0.10. This represents a pattern where effect sizes with *p*-values in the first interval ($p < 0.01$) have a weight of 1.00, or are 100% likely to survive selection, while effect sizes in the second interval are only 90% likely to survive,

and so on. These weights apply to the effect sizes in the baseline group, here referred to as "Group One." λ is free to vary; assume that it is estimated as 0.50. Recall that λ applies only to nonsignificant intervals, where $p > 0.025$; in other words, it applies to the third, fourth, and fifth weights. Therefore, the model has estimated weights of 1.00, 0.90, 0.30, 0.20, and 0.05 for the second group, "Group Two." In this hypothetical case, where a strong one-tailed selection pattern is proposed by the *a priori* weights, the model indicates that survival of nonsignificant effect sizes is even less likely in the second group. The researcher has gained information about the differences in bias patterns across groups while estimating only one parameter.

The lambda model for sensitivity analysis shares the redeeming features of the original Vevea and Woods (2005) model, although it does possess a few limitations. It can include a variance component and moderators, meaning that it can still accommodate both systematic and random heterogeneity. It can function with small datasets. However, it operates under the same assumptions as the original lambda model – namely, it assumes that the meta-analyst has correctly identified the relevant study characteristic and assigned the effect sizes to groups. It also assumes that the differences in weights across groups can be represented by a multiplicative constant affecting the nonsignificant weights.

Using the lambda model as a sensitivity analysis does require the meta-analyst to select some patterns of *p*-value intervals and weights. There is no "wrong" pattern to implement; the meta-analyst can experiment and assess the effects of various patterns of publication bias on their dataset. However, the meta-analyst should take care to try out a range of bias patterns, to be satisfied that the data are robust to different selection patterns. The number of intervals is no longer an issue, so the meta-analyst can try out one-tailed and two-tailed selection patterns with any given severity and can include steps at any cutpoint that may be psychologically relevant.

In the next two sections, respectively, I demonstrate the use of the lambda model as a sensitivity analysis on the Bem et al. (2016) dataset and go on to explore its performance through simulation.

5.1 Example

For the substantive example, I implement five selection-bias patterns. Four of these patterns match the ones described in Vevea and Woods (2005). I strongly emphasize that it is not necessary to use these specific patterns; they were not intended to be a rule of thumb, merely an example. Because this dataset is featured only for illustration purposes, though, I include them here. Again, substantive meta-analysts are strongly encouraged to experiment with different selection patterns, and to implement patterns other than those presented in Vevea and Woods (2005). To this end, I include a fifth bias pattern of my own.

These five selection-bias patterns are presented in Table 4. Note that, for one-tailed selection, the weights are larger when the *p*-values are smaller, corresponding to one-tailed alpha levels. For two-tailed selection, weights are larger for the smallest and largest *p*-values, corresponding to two-tailed alpha levels. I refer to the fifth pattern as "extreme two-tailed selection." It describes a situation in which significant effect sizes (those with $p < .05$ or $p > .95$) always survive selection, while nonsignificant effect sizes

are much less likely to survive, with probabilities as low as .10 for *p*-values between .25 and .75. In a sense, this fifth pattern demonstrates much stronger two-tailed selection than the fourth pattern, where probabilities drop only to .25.

Table 4. The selection-bias patterns demonstrated on the Bem dataset.

***p*-value interval**	**(1) Moderate one-tailed selection**	**(2) Severe one-tailed selection**	**(3) Moderate two-tailed selection**	**(4) Severe two-tailed selection**	**(5) Extreme two-tailed selection**
.000-.005	1.00	1.00	1.00	1.00	1.00
.005-.010	.99	.99	.99	.99	1.00
.010-.050	.95	.90	.95	.90	1.00
.050-.100	.90	.75	.90	.75	.50
.100-.250	.80	.60	.80	.60	.25
.250-.350	.75	.50	.75	.50	.10
.350-.500	.65	.40	.60	.25	.10
.500-.650	.60	.35	.60	.25	.10
.650-.750	.55	.30	.75	.50	.10
.750-.900	.50	.25	.80	.60	.25
.900-.950	.50	.10	.90	.75	.50
.950-.990	.50	.10	.95	.90	1.00
.990-.995	.50	.10	.99	.99	1.00
.995-1.000	.50	.10	1.00	1.00	1.00

The results of these analyses are presented in Table 5, which shows the adjusted mean and variance-component estimates for each selection pattern, along with λ. Remember that λ is not fixed, which allows us to explore the differences in bias patterns across groups. An alternative version (fixing λ) would allow us to explore different fixed values of λ but would ultimately be less informative.

Table 5. The results of the lambda model sensitivity analyses on the Bem data.

Parameter	**Pattern (1)**	**Pattern (2)**	**Pattern (3)**	**Pattern (4)**	**Pattern (5)**
Variance Component (τ^2)	0.0018	0.0039	0.0000*	0.0000*	0.0000*
Intercept (β_0)	0.0338	-0.0047	0.0384	0.0331	0.0255
Lambda (λ)	0.3299	0.3742	0.2729	0.3606	0.7615

Notes: * indicates that a border condition was present, so the fixed-effect estimates are provided. (They equal the random-effects estimates, as the variance component is essentially zero.)

5.1.1 The Intercept

First, we can look at the intercept estimates. This allows us to assess the effect of the various selection-bias patterns on the overall mean. The adjusted intercepts will almost certainly vary across bias patterns; therefore, the question is not whether variance among them exists but about the magnitude of their variability. For this example, remember that the unadjusted mean effect, or intercept, is 0.0714 (see Table 2) – in other words, that participants who are presented with retroactive stimuli will remember more words than their counterparts, performing better by approximately 0.07 standard deviations.

Before we look at the variability of these mean estimates, we should discuss the difference between the concepts of statistical significance and clinical significance. An effect size may be statistically significant, even at a stringent alpha level, but may still be too small to be of practical or substantive interest. Hypothetically, a researcher with a sufficiently large sample size might find even the most minute effect sizes statistically significant. However, the clinical significance of the effect also matters. Consider the unadjusted estimate. Is an increase of 0.07 standard deviations in the number of words recalled practically, or clinically, important? Precognitive research is somewhat unique in that researchers are often interested in any effect size, regardless of its magnitude. If an infinitesimal effect exists, psi researchers proclaim that there is still evidence of precognition, albeit small. In this case, although 0.07 standard deviations is a small effect, it is on the large side for precognition. My goal is not to campaign either for or against the existence of precognitive powers, but I invite readers to think carefully about what they are willing to consider clinically significant.

The most severe bias patterns are Pattern #2 (dubbed "severe one-tailed selection") and Pattern #5 ("extreme two-tailed selection"). As one might expect, those patterns are the ones which yield the smallest adjusted effect sizes. For Pattern #2, the adjusted effect is 0.0032 – virtually indistinguishable from zero, even in the field of precognition. For Pattern #5, the adjusted effect is 0.0340, a reduction of approximately 48%; the effect is still present but has been drastically reduced.

It is important to remember that these adjusted estimates are the result of artificial bias patterns I have imposed on the data. The bias patterns are subjective and can be changed at will. Therefore, I would no more accept, say, 0.0032 as the one true effect-size estimate than I would the original 0.0714. These estimates simply represent what the adjusted effect would be, given a certain precise selection-bias pattern. The question of interest here is whether bias patterns of varying severity are capable of drastically reducing the effect size. In this case, it certainly seems so. This indicates that the effect is not robust to selection, and that it may be an artifact of publication bias. (Again, this assumes that we are willing to accept the presence of an effect to begin with.)

5.1.2 Lambda

Next, we can look at the estimates of λ. Across all five selection patterns, λ remains below one, indicating that the nonsignificant weights are smaller for the group coded 1 (the studies published before 2011). This is evidence that the direction of the

difference in bias patterns is consistent regardless of the fixed bias pattern imposed upon the group coded 0.

The estimate of λ hovers around 0.30, apart from Pattern #5, which is the most extreme weight pattern. The different magnitude of λ for Pattern #5 is likely because λ is a multiplicative constant on the nonsignificant weights. As the number of nonsignificant weights changes, along with their fixed values, the estimate of λ will typically change as well.

Much like the intercept estimates, these values of λ are the result of subjective fixed patterns. They were not estimated from the data and should not be interpreted as the true population value of λ. The question of interest is whether the difference between groups appears to hold constant across bias patterns, and in this case, it does. Pattern #5 is the exception – even in that situation, however, the direction of the difference is consistent.

In the next section, I evaluate the performance of the lambda model as sensitivity analysis through simulation.

5.2 Simulation Results

Again, for each cell, I conducted approximately 10,000 replications.

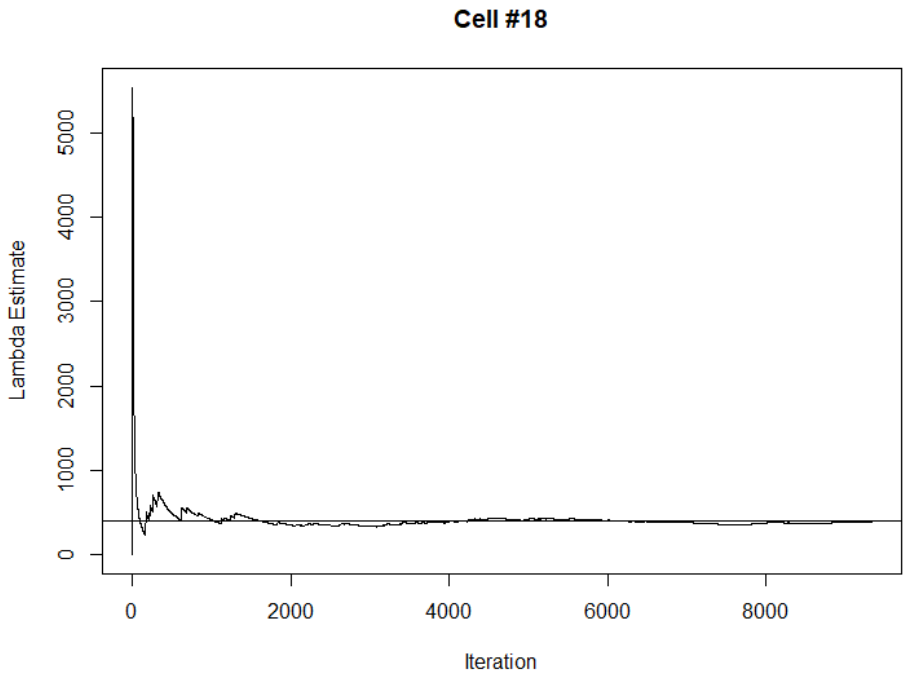
I estimated the lambda model as a sensitivity analysis twice, each time with a different set of fixed weights. The two sets of fixed weights correspond to Set 1 and Set 3 in the substantive example section described above; that is, the first set is an example of moderate one-tailed selection, and the second set an example of moderate two-tailed selection.

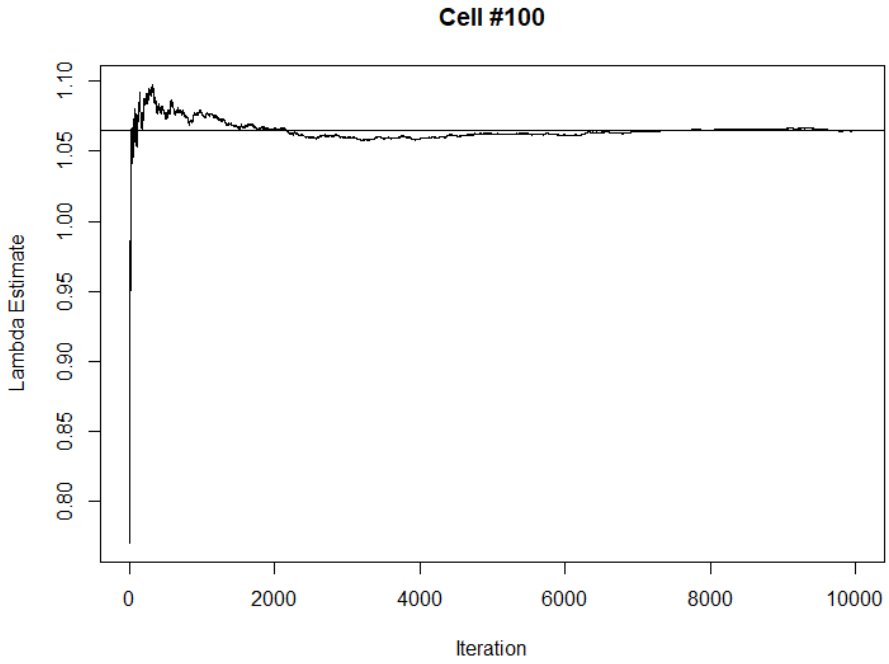
I now present assessments of simulation convergence, followed by the simulation results. The results in this chapter will be shorter than those presented in Chapter 4 for one simple reason; the mean and variance component are estimated based on user-specified parameters, so it is not meaningful to assess their accuracy. I focus on the estimate of λ, which conveys information about the bias pattern in the second group relative to the user-specified bias pattern. I present plots of the parameter estimates and discuss the model's performance.

5.2.1 Convergence

The following plots show the cumulative average λ estimates, like those described at the end of Chapter 4. Again, we cannot absolutely prove that the estimates have converged; we aim to demonstrate that there is no evidence of non-convergence.

Plots of the cumulative mean estimate for four cells are presented in Figure 25. These plots are all from selection pattern 1, but the results do not vary across selection patterns.


Cell #18
Lambda Estimate
0
1000
2000
3000
4000
5000
0
2000
4000
6000
8000
Iteration


Cell #100
Lambda Estimate
0.80
0.85
0.90
0.95
1.00
1.05
1.10
0
2000
4000
6000
8000
10000
Iteration

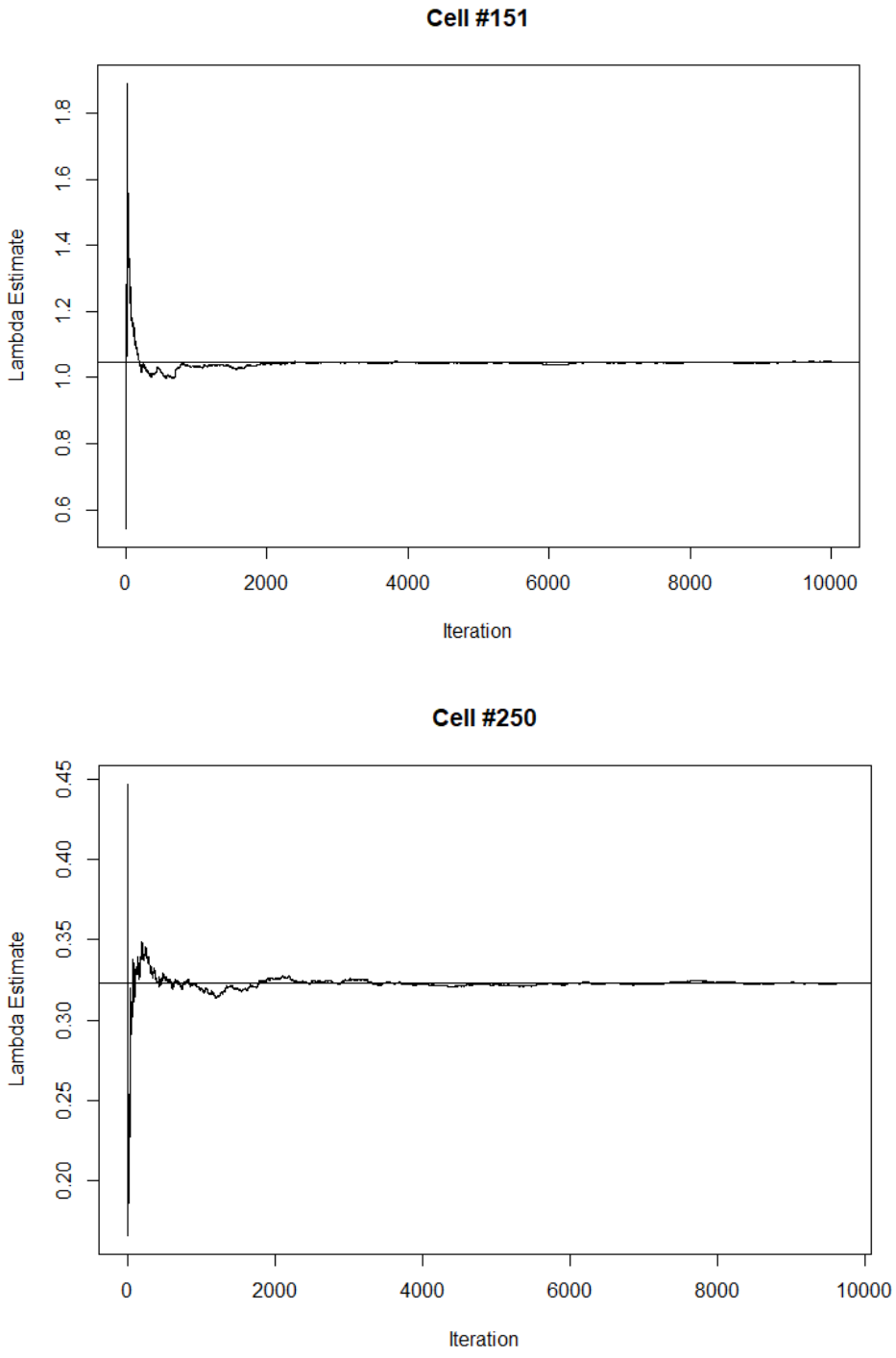


Figure 25. Plots of the cumulative mean for the lambda model as sensitivity analysis.

As before, the results do not demonstrate evidence of non-convergence, so we can assume that the average parameter estimate across iterations is a valid estimate.

5.2.2 λ Estimate

As described in the simulation section at the end of Chapter 4, the plots of the average estimates of λ are slightly more complicated. While the population mean is the same across all cells and all conditions (0.20), λ is an estimate of the difference in bias patterns across groups, as a multiplicative constant. This means that the "true" estimate of λ varies across levels of bias pattern. I invite the reader to refer to the end of Chapter 4 for a detailed explanation.

The gray lines on these plots still represent the approximate predicted patterns of λ estimates, while the black lines represent the observed patterns. The more the gray and black dots (or lines) overlap, the more accurate the reproduction of λ. It is important to note that the λ estimates reported here are likely to be less accurate than those reported in Chapter 4. The reason for this is because, by specifying a fixed set of weights, the model we are estimating no longer approximates the data generating pattern (which generally simulates weights for two intervals, one significant and one nonsignificant). λ must therefore compensate for this mismatch, and the results are liable to be inaccurate, particularly for the second set of weights (the lambda model recognizes all *p* values greater than .05 as nonsignificant and does not by default understand a two-tailed alpha level of .95).

Such inaccuracy is not much of a concern in this context, however, because meta-analysts are using these results only as a sensitivity analysis and will be more interested in the general magnitude of λ than in its precise value.

Figure 26 displays the results for cells with I^2 of 0%, using the first set of fixed weights (representing moderate one-tailed selection). The lines now represent bias patterns, rather than models; the solid line depicts the cases where no bias is present in one group and weak bias is present in the other, and so on.

In the top left panel of Figure 26, the estimate of λ is fairly accurate for the "None vs. Strong" and "Weak vs. Strong" bias patterns when *k* is 48 and 172. By *k* of 172, even the estimate of "None vs. Weak" is approaching the population value. The same general pattern holds in the bottom left and bottom right, although the estimates are less accurate. The "None vs. Strong" and "Weak vs. Strong" bias patterns in the top right panel, where bias is generated as an exponential function of *p*-values, yield the most inflated estimates.

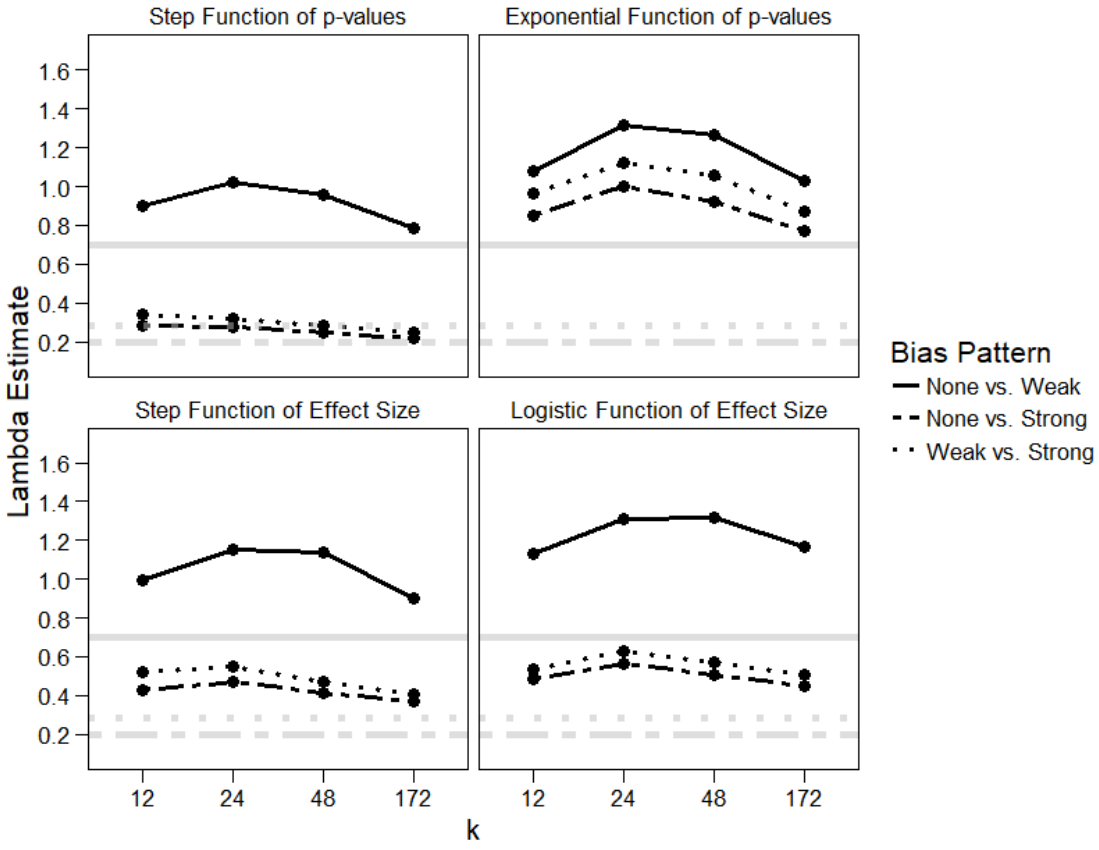


Figure 26. Estimates of λ across methods of bias generation and bias pattern from cells with I^2 of 0%, using selection pattern 1.

Figure 27 displays the results for cells with I^2 of 75%, using the first set of fixed weights.

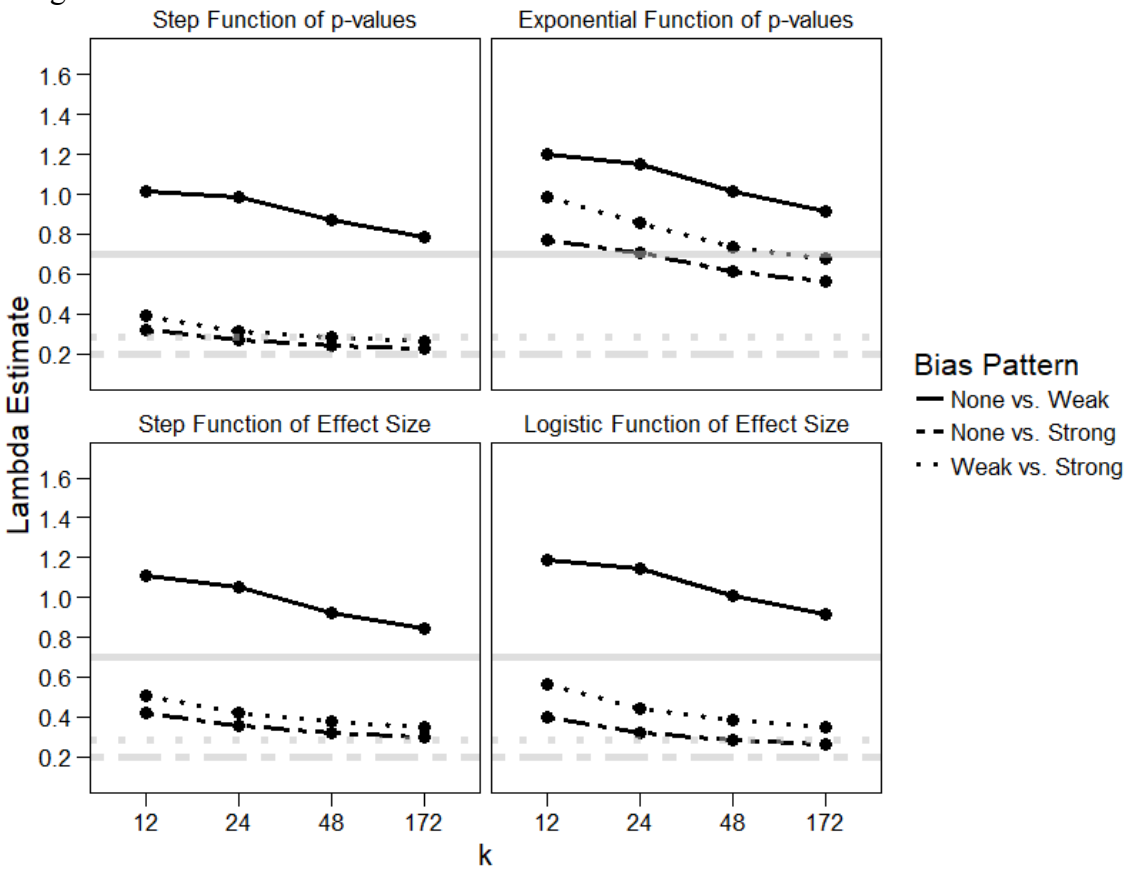


Figure 27. Estimates of λ across methods of bias generation and bias pattern from cells with I^2 of 75%, using selection pattern 1.

The same overall pattern holds in Figure 27 as it does in Figure 26, with the note that, in general, the estimates of λ become more accurate in the presence of some heterogeneity, and they are still fairly accurate even across methods of bias generation.

We now move on to the results from the second selection pattern. Figure 28 displays the average estimates of λ for cells with I^2 of 0%, using the second set of fixed weights (representing moderate two-tailed selection bias).

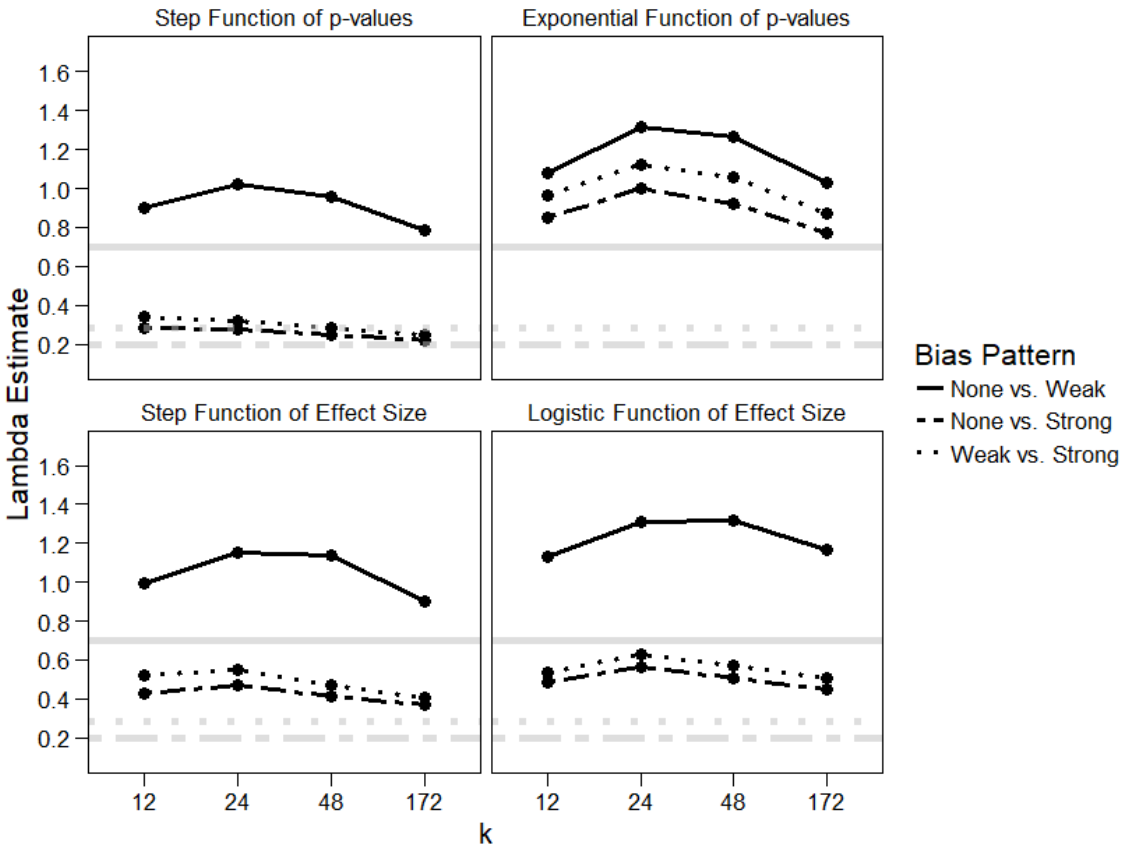


Figure 28. Estimates of λ across methods of bias generation and bias pattern from cells with I^2 of 0%, using selection pattern 2.

Figure 29 displays the results for cells with I^2 of 75%, using the second set of fixed weights.

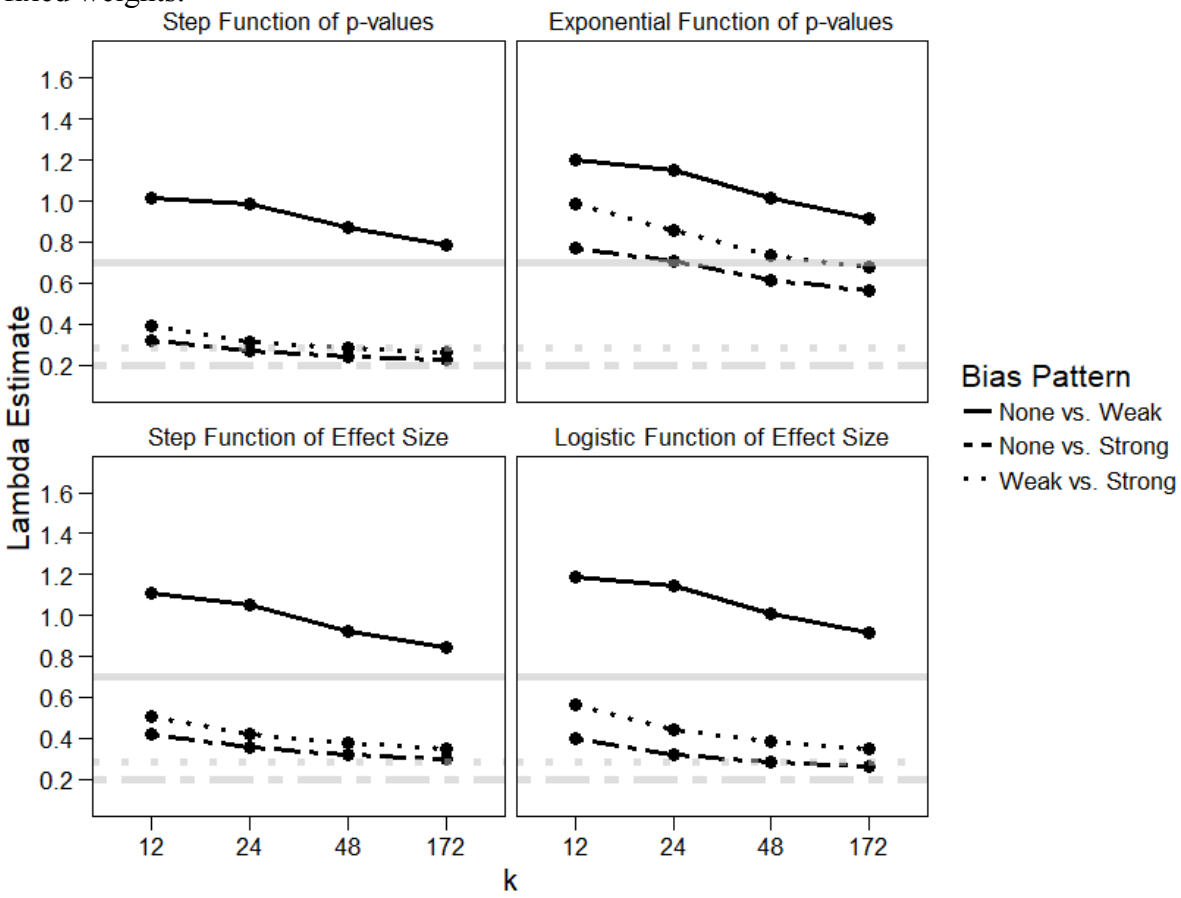


Figure 29. Estimates of λ across methods of bias generation and bias pattern from cells with I^2 of 75%, using selection pattern 2.

If heterogeneity is present (and high), the model is surprisingly robust to violations of its assumptions. Even when a set of weights has been specified that directly conflicts with the model's assumptions, it does a relatively good job of reproducing the predicted values. (Of note, however, is the fact that it still performs most poorly in the top right panel, the case where bias is generated as an exponential function of *p*-value. That specific generation mechanism consistently results in the least accurate estimates.)

5.3 Conclusions

This chapter presents a version of the lambda model that can work with a fixed set of weights and estimating only λ. Doing so allows interested meta-analysts to implement the lambda model on empirical datasets, regardless of the sample size of said datasets. In the process, of course, the researcher sacrifices the ability to rely on the adjusted parameter estimates, as they are no longer being estimated from the observed data. He or she, however, gains the opportunity to test out a wide range of selection patterns on their data, and to learn about the differences in bias patterns across groups in the process.

The simulation presented in this chapter yields promising results. Even in cases where the fixed set of weights represents a pattern of two-tailed selection (a condition that the lambda model, in its present form, cannot yet handle correctly), even when selection bias is generated in a way that violates the model assumptions, and even in the presence of high heterogeneity, the lambda model generally does a good job of approximating the predicted parameter value.

In future research, it may be worthwhile to extend this simulation to different sets of fixed weights. Implementing this version of the lambda model on additional substantive datasets, such as those presented in Coburn and Vevea (2015), would also be productive. For now, however, we proceed to Chapter 6 – a Bayesian implementation of the lambda model.

Chapter 6: Bayesian Adaptations with *R* and *JAGS*

Previously, all discussions of the lambda model have referred to it in the context of maximum-likelihood estimation. This estimation method has several useful properties. Under certain circumstances and as the sample size approaches infinity, maximum likelihood yields consistent, efficient, asymptotically normal, and minimum variance estimates (Scholz, 1985). However, with finite sample sizes, maximum-likelihood estimation is not always ideal. The lambda model often involves estimating many parameters relative to a small dataset. Using maximum likelihood, the meta-analyst's best hope of obtaining model convergence (and reasonable parameter values) is to reduce the number of parameters. For the smaller datasets, reducing the number of parameters may mean eliminating most *p*-value cutpoints, perhaps leaving only one (often at $p = 0.05$). In these cases, the model is still a useful tool, of course – but it would certainly be more useful if more weights were estimated. Therefore, it is advantageous to estimate the model using a method that is not so easily affected by sample size.

Using maximum likelihood to estimate the lambda model poses an additional complication. To ensure that the model is identified, at least one parameter must be fixed; this is accomplished by constraining the weight for the first *p*-value interval equal to 1 and interpreting the weights for subsequent intervals relative to it. In other words, a weight of 0.50 indicates that effect sizes with *p*-values in that interval are half as likely to survive selection as those with *p*-values in the first interval. In the maximum-likelihood context, a weight of 0.50 does not mean that effect sizes in that interval have a 50% chance of survival. This example is somewhat intuitive; however, if subsequent weights were estimated at 0.76, 1.23, and 0.41, understanding and interpreting the results can quickly become difficult.

Bayesian estimation is an ideal candidate for handling both issues. It is not based on large sample theory, and therefore large samples are not required to obtain convergence (van de Schoot & Depaoli, 2014). A body of work exists, including but not limited to simulation studies, demonstrating the benefits of Bayesian statistics in the context of small data sets (e.g., van de Schoot & Depaoli, 2014; Zhang et al., 2007). Additionally, in Bayesian statistics, parameter estimates are obtained from a posterior distribution, which is formed from the combination of the likelihood given the data and the prior distribution. Rather than constraining the first weight to one, the meta-analyst can specify a relevant prior distribution, i.e. $U(0, 1)$. As the meta-analyst goes on to specify priors – typically $U(0, 1)$ – for the subsequent weights, these weights still cannot be interpreted as probabilities (because it is impossible to know the number of effect sizes that existed in each interval prior to selection), but they can be interpreted relative to each other, rather than only to the first weight. For example, consider using Bayesian estimation with *p*-value cutpoints at 0.05 and 0.10. This creates three intervals: $p < 0.05$, $0.05 < p < 0.10$, and $0.10 < p < 1.00$. Assume that the model yields weight estimates for these intervals of 0.80, 0.50, and 0.20 (respectively). These can now be interpreted relative to each other. In a Bayesian context, the meta-analyst can make statements like, "Effect sizes with *p*-values between 0.10 and 1.00 are 40% less likely to survive selection than those with *p*-values between 0.05 and 0.10." (Note that 0.40 = 0.20/0.50.)

This interpretation is more intuitive and allows the meta-analyst to compare all intervals, rather than being forced to compare each weight to the first. Being able to interpret the model results in this way also makes the model more user-friendly and approachable. For these reasons, among the others presented above, the lambda model is an ideal candidate for Bayesian estimation.

6.1 Implementing the Lambda Model

I have created a Bayesian version of the lambda model using *R* (R Core Team, 2017) and JAGS, or Just Another Gibbs Sampler (Plummer, 2003). The *R* package *R2jags* (Yu-Sung & Masanao, 2015) provides a wrapper for JAGS in the *R* interface, making it easy to estimate models and run simulations through the *R* console. Because the density of the lambda model is not pre-specified in JAGS, the way that common densities are (e.g. the normal density), I employed a method called the "zeroes trick" (Lund, Jackson, Best, Thomas, & Spiegelhalter, 2013) to code and implement the model.

The zeroes trick relies on the fact that a Poisson observation of zero, or $\phi(0)$, has a likelihood of $e^{-\lambda}$.[5] Setting λ equal to the negative log of the desired arbitrary likelihood, or $-\log L_i$ (where i is an index of sample size) and specifying a set of zeroes as observed data yield the desired likelihood contribution. It is important to keep in mind, however, that λ must ultimately be positive because it is the mean of the Poisson distribution; therefore, the zeroes trick occasionally requires adding a constant to the negative log-likelihood. The value of the constant does not matter, as long as it is sufficiently large to ensure λ remains positive.

Specification of prior distributions is a crucial aspect of Bayesian estimation. Priors must be specified for every parameter that is estimated in the model. In the case of the lambda model, this means priors must be specified for the mean (or intercept) and any conditional means, the variance component, λ, and weights for the *p*-value intervals. The ability to specify a prior distribution incorporating one's own knowledge about the parameters involved is one of the premier benefits of Bayesian estimation. However, poorly or incorrectly specified priors may seriously impact the resulting posterior distributions and parameter estimates (for an example, see Depaoli, 2014), and research exploring so-called non-informative priors indicates that such priors may be more informative than believed (Gelman, 1996, 2006, 2009). As a compromise, some authors advocate weakly-informative priors, which attempt to remain vague while still restricting the parameter space to plausible values (Gelman, 2009); this can be especially helpful with complex models.

I used the same set of priors for all the Bayesian analyses presented in this chapter. For the mean, or β_0, I specified a prior of N(0, 0.00001^2); this yields a distribution centered at zero with precision that is also essentially zero (0.0000000001), or a very, very wide normal distribution centered at zero. The prior for the inverse variance component, $1/\tau^2$, is $\Gamma(0.001, 0.001)$, or a gamma distribution with shape (α)

[5] λ is absolutely standard notation for the Poisson distribution. In this paragraph only, use of λ is specific to the Poisson context. In all other cases, λ refers to the corresponding parameter of the lambda model.

and rate (β) parameters set to 0.001. Almost all the mass in such a prior distribution is near zero. The gamma distribution is a common conjugate prior in Bayesian estimation, and is often used as a prior for measures of precision (the inverse of a variance). It is used here for the same reason; precision cannot be negative and is positively skewed. There are no moderators in these simulations, so the remaining parameters are the weights and λ. The priors for all the weights are U(0, 1) – a uniform distribution from zero to one. The weights are bounded in this way so that, although they are not technically probabilities, they can roughly be interpreted as such. The prior for λ is U(0, 100). This is done in part to mirror the priors for the weights and in part because λ cannot be negative. The upper bound of the prior is set to 100; in practice, it is possible, although highly unlikely, that λ could exceed 100, but for the purposes of this dissertation, such a situation did not arise. Setting the upper bound at 100 will ideally aid model estimation by constraining the results to a reasonable range.

Bayesian analysis, which originally required (often difficult) analytical integration or approximation, can now be conducted much more easily using Markov chain Monte Carlo methods (Krushke, Aguinis, & Joo, 2012). The principle of Monte Carlo integration is that one can approximate a given posterior distribution using a large representative random sample of parameter values drawn from said distribution (Gilks, Richardson, & Spiegelhalter, 1996). From this large sample, the user can calculate the posterior mean, quantiles, shape, and so on (Krushke, Aguinis, & Joo, 2012). Rather than needing to compute complicated integrals, Monte Carlo integration allows us to generate a sample of parameter values simply by specifying a prior distribution and the form of the relevant likelihood function. Markov chains are used to handle the sampling procedure. In a Markov chain, the next sample drawn from a given distribution depends only on the current state of the chain – that is, on the current sample (Gilks, Richardson, & Spiegelhalter, 1996). Assuming that the chain does not start in a wildly inappropriate location (in other words, that its initial values are reasonable), and subject to certain terms and regulatory conditions, the Markov chain will eventually "forget" its initial state and converge on a unique posterior distribution (Gilks, Richardson, & Spiegelhalter, 1996). When the chain begins, however, each sample, or state, will depend more highly on the previous states and therefore on the initial values. Averaging across all states, or samples, in the chain without taking this early dependence into account would bias the posterior parameter estimates. As a result, analysts usually set aside a certain number of early samples as a "burn-in period" to be discarded. The length of the required burn-in period often varies depending on several factors, including the complexity of the model and the specific initial values; because of this, it is beneficial to conduct a trial run and determine the burn-in period based on the chain's behavior.

In some cases, it may be difficult for a model to converge on a unique posterior distribution. If each new sample (or state) of a chain depends too highly on the previous sample, some autocorrelation is present. Allowing the chain to run for a longer period can often eliminate this problem. However, with Bayesian estimation, computational power may be at a premium, and longer run times are not always practical. In this case, or if data storage is limited and the user physically cannot store every iteration of the chain, a "thinning" parameter may be used. Thinning a chain results in discarding iterations, much like the burn-in process – although, rather than discarding a block of iterations at the start, thinning maintains every n^{th} iteration. That is, if the thinning parameter is set to 5,

the first four iterations are discarded, the fifth is maintained, and so on. If the thinning parameter is 1, all iterations are stored. Of course, different researchers have different opinions of thinning. In this dissertation, I always set the thinning parameter to 1 unless otherwise specified. I saved all iterations of all replications in all simulation cells.

It is also advisable to run multiple Markov chains, rather than just one. Views on this issue have also been conflicting, with recommendations ranging from many short chains to a few long ones or to one extremely long chain (Gelfand and Smith, 1990; Gelman and Rubin, 1992a, b; Geyer, 1992; Gelman, 1995). If multiple chains are estimated, it is important to determine that convergence has occurred not only within the chains but also across the chains. With more complicated models, particularly mixture models, other complications such as label switching may arise (Jasra, Holmes, & Stephens, 2005). Overall, running multiple long chains is generally worthwhile if possible (Gilks, Richardson, & Spiegelhalter, 1996); if one long chain seems to have converged but there is no convergence across chains, this may lead to an important investigation of the model or initial values. The researcher may also use a different set of initial values for each chain to ensure that their results are not dependent on the initial values, providing additional support for model convergence (Gelman and Rubin, 1992a, b; Gelman, 1995).

One also must determine when to stop the chains – that is, how long the chains should be, or how many iterations they will consist of. The chains should run long enough for their results to converge; if too much autocorrelation is present, the chains should be run longer. For simple models, the chains may converge after a relatively low number of iterations. Much like the issue of sample size in substantive research, more iterations are almost always better; however, also much like the sample size issue, the longer a chain the more expensive it is (in terms of computing power and storage space). Of course, once the chain has completely converged, additional iterations are not likely to change the value of the posterior estimates drastically. The difficulty lies in determining how long is "long enough." Finally, the researcher must also decide on some set(s) of initial values. With multiple chains, it is useful to give each chain a unique set of initial values; this allows one to assess the impact of any given starting position, and convergence across the chains in this case will reinforce one's confidence in the posterior parameter estimates.

Bayesian estimation certainly requires a lot of decisions on the part of the researcher (more so than maximum-likelihood estimation). It is complicated, and sometimes decisions that seem small can have an unforeseen impact on the results. If thoroughly researched and carefully conducted, however, Bayesian estimation yields many benefits.

The next section presents the results of the Bayesian lambda model, using the substantive dataset described previously. Finally, the last section of this chapter presents the results of a simulation exploring the performance of the Bayesian lambda model.

6.2 Example

For the Bem et al. (2016) dataset, I estimated the Bayesian lambda model with six *p*-value cutpoints, resulting in a total of six estimated weights (as the first weight now does not need to be fixed to one). Estimating so many intervals in a maximum-likelihood

context is often difficult, depending upon the size of the dataset. However, Bayesian estimation handles the problem easily.

The *p*-value cutpoints I specified are $p = 0.025, 0.05, 0.10, 0.50, 0.90$, and 1.00; these match the cutpoints specified for the example in Chapter 4. I generated three chains, each with 5,000 burn-in iterations and each retaining 15,000 iterations post-burn-in, and each with its own unique (automatically generated) set of initial values. I specified the other information (regarding priors and thinning) as described above. I coded the earlier studies (published before 2011) as 1's and the later studies (published after 2011) as 0's.

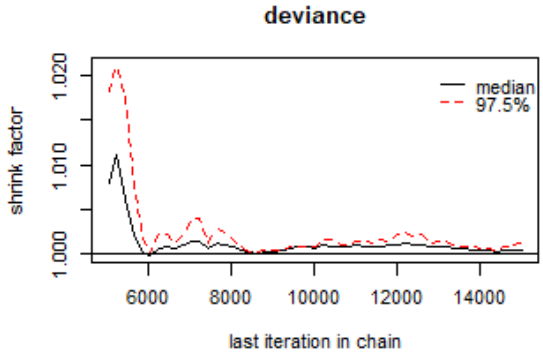
6.2.1 Model Convergence

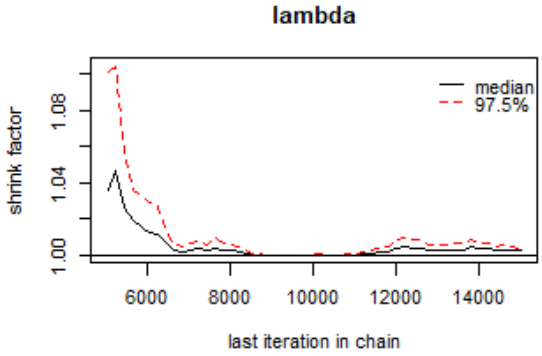
Before examining the model parameter estimates, it is important to assess model convergence. It is impossible to prove convergence, but it is possible to fail to provide evidence of non-convergence. There are many convergence diagnostics for Bayesian estimation, both within-chains and between-chains. Several of these diagnostics are included in the *R2jags* package (Yu-Sung & Masanao, 2015). Gelman (2004) provides a useful discussion of convergence diagnostics.

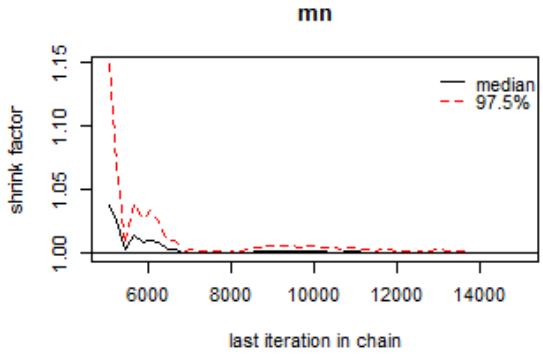
The first method I used is the Gelman-Rubin (1992) convergence diagnostic. It compares the variance in parameter estimates between chains to the variance in the estimates within chains, essentially by conducting an analysis of variance (ANOVA) (Gelman & Rubin, 1992). If there is a significance difference between the variances, this indicates that the chains have not converged; that is, that there is a difference between chains, which can only be due to their differing initial values (Gelman & Rubin, 1992). Ideally, the chains have been allowed to run for so many iterations that they "forget" where they started (i.e., their initial parameter values), and their starting places will have no impact on their variance. The ANOVAs will be nonsignificant and the test statistic for each parameter, which follows an *F*-distribution, will be approximately 1. If the test statistics are greater than 1 and there is a difference between the chains, running the chains for longer would likely eliminate this difference and reduce the impact of the starting values by the scale of the test statistic. Therefore, the test statistics are called the "potential scale reduction factors" (PSRF).

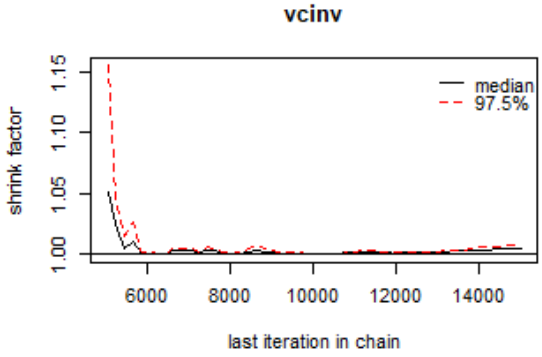
For this dataset, there are ten parameters – the mean, the variance component, λ, and seven weights. I ran the Gelman and Rubin (1992) diagnostic with the function *gelman.diag( )*. All nine parameters, including the deviance parameter, yielded a PSRF of exactly 1.00. Guidelines indicate that PSRF below 1.10 suggest adequate convergence (Gelman & Rubin, 1992); therefore, all parameters appear to show no signs of non-convergence across all three chains.

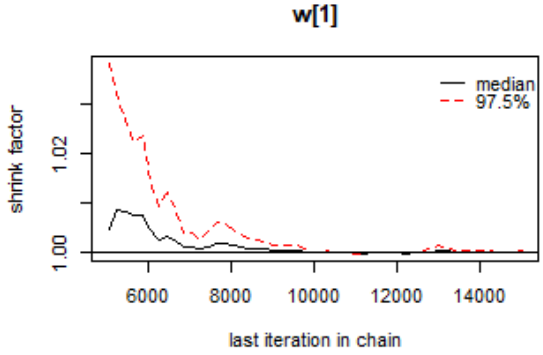
Figure 30 shows plots of the development of the scale reduction factors for each parameter across the chain iterations. These indicate whether the chains may be stable at reduced numbers of iterations. Although, for the Bem et al. (2016) dataset, the chains may be stable before reaching 15,000 iterations, data storage is not remotely an issue, so there is no reason to reduce the number of iterations.

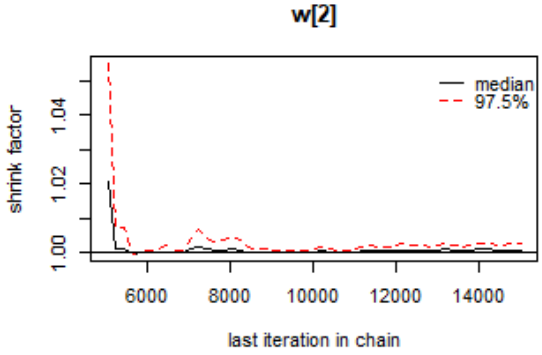

deviance
shrink factor
1.000
1.010
1.020
median
97.5%
6000
8000
10000
12000
14000
last iteration in chain


lambda
shrink factor
1.00
1.04
1.08
median
97.5%
6000
8000
10000
12000
14000
last iteration in chain


mn
shrink factor
1.00
1.05
1.10
1.15
median
97.5%
6000
8000
10000
12000
14000
last iteration in chain


vcinv
shrink factor
1.00
1.05
1.10
1.15
median
97.5%
6000
8000
10000
12000
14000
last iteration in chain


w[1]
shrink factor
1.00
1.02
median
97.5%
6000
8000
10000
12000
14000
last iteration in chain


w[2]
shrink factor
1.00
1.02
1.04
median
97.5%
6000
8000
10000
12000
14000
last iteration in chain

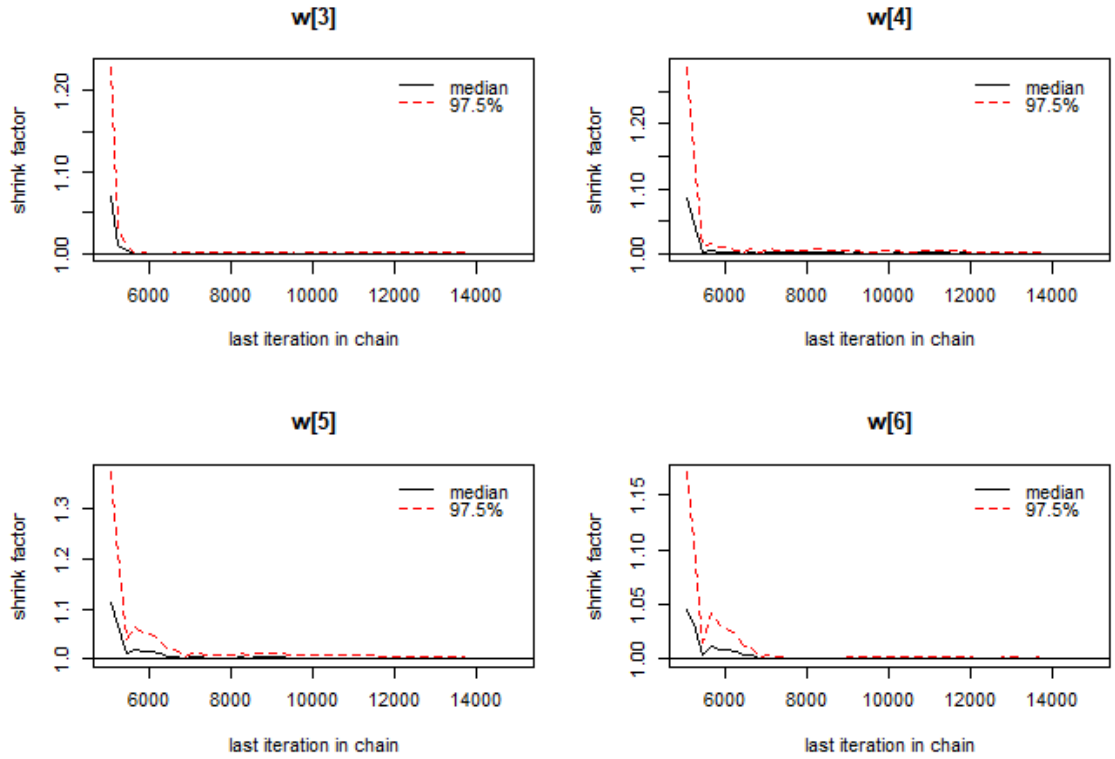


Figure 30. Plots of the development of scale reduction factors for each parameter.

The Geweke (1992) diagnostic is also useful. It is calculated once per chain, and it is based on a test for the equality of the means of the first and last parts of a given Markov chain. By default, the "first" part is defined as the beginning 10% of the post-burn-in iterations, and the "last" part as the final 50% (Geweke, 1992), although these proportions can be manually altered. If the means of these two parts are equal for each chain, there is no evidence of non-convergence. The test statistic for the Geweke (1992) diagnostic follows an asymptotically standard normal distribution. If the test statistic is significant for any parameter within a chain, that indicates that said parameter has not reached convergence.

I ran the Geweke (1992) diagnostic using *geweke.diag( )* and calculated the *p*-values for each test statistic. For the first chain, the means of the two chain segments did not differ significantly for any of the parameter estimates (all $p > 0.05$). For the second chain, two parameters (the fourth and fifth weights) reached significance ($p < .05$). For the third chain, none of the parameters were significant ($p > .05$). Examining the other convergence diagnostics and the actual trace plots for each parameter will determine whether this is a concern.

The Heidelberger-Welch (Heidelberger & Welch, 1981; 1983) diagnostic test uses the Cramer-von-Mises statistic to test the null hypothesis that the parameter values for each individual chain come from a stationary distribution. It is applied first to the entire chain; if the null hypothesis is rejected, indicating that the parameter distribution is not stationary, the first 10% of the chain is discarded and the test conducted again. This procedure continues until the first 50% of the chain is discarded; if the null hypothesis is

still rejected, this constitutes "failure" of the test, or evidence of non-convergence, and indicates that more iterations are needed.

To run this and the upcoming final diagnostic test, I installed the *superdiag* (Tsai, Gill, & Rapkin, 2015) *R* package and used the *superdiag( )* function. This function also automatically calculates the Geweke (1992) and Gelman and Rubin (1992) diagnostics as well. All parameters, for all chains, passed the Heidelberger-Welch test for a stationary start. In addition, all parameters (again for all chains) passed the Heidelberger-Welch halfwidth test. Neither of these tests presents any evidence of non-convergence.

Finally, I implemented an MCMC diagnostic that aims to determine the number of iterations required to reach a given level of precision for each parameter estimate (proposed by Raftery & Banfield, 1991; Raftery & Lewis, 1992). The goal of this diagnostic is to determine the minimum number of burn-in (defined as M) and post-burn-in (N) iterations required, as well as the minimum thinning interval (k), to reach ideal precision. Precision is defined by a preset quantile of interest (e.g., 0.025, 0.50, etc.), degree of accuracy, and probability. The diagnostic tool then produces a lower-bound value (minimum number) for M, N, and k based upon these preset criteria.

For the Raftery and Lewis (1992) diagnostic, the *superdiag( )* function automatically varies the specified quantile (q), accuracy (r), and probability (s) per chain. For Chain 1, $q = 0.001$, $r = 0.005$, and $s = 0.95$; the diagnostic produced a minimum of 3,746 iterations post-burn-in. For Chain 2, $q = 0.1$, $r = 0.005$, and $s = 0.90$, and the diagnostic produced a minimum of 9,740 post-burn-in iterations. Finally, for Chain 3, $q = 0.05$, $r = 0.005$, and $s = 0.999$; in this case, the diagnostic produced a minimum number of 20,573 post-burn-in iterations. For all three chains, the model was estimated with 15,000 post-burn-in iterations, so it is likely that no more iterations are necessary to achieve the specified levels of precision – at least according to the Raftery and Lewis (1992) diagnostic. (Regarding the third chain, I will tentatively assume that 15,000 are enough, keeping the other evidence in mind and knowing that this diagnostic can be overly optimistic in terms of the minimum number of iterations.)

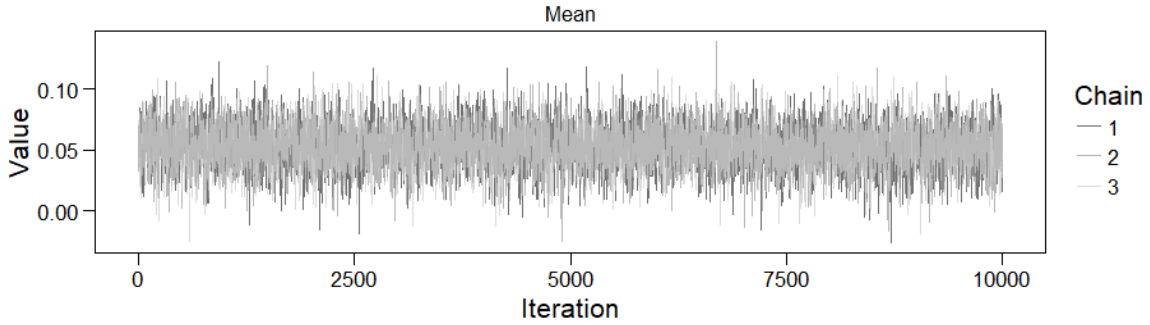
These convergence diagnostics, of course, are not without flaws, and a conscientious researcher should not base their conclusion regarding convergence or non-convergence solely on the results of these statistical tests. For instance, the Geweke (1992) and Heidelberger-Welch (1981; 1983) diagnostics are conducted once per parameter per chain, resulting in a total of sixty statistical tests and an inflated Type I error rate. It is also possible that these diagnostics are wrong – a test may not be significant, but evidence of non-convergence may still be present. With that in mind, we proceed to examine trace plots of the three chains for each of the ten parameters.

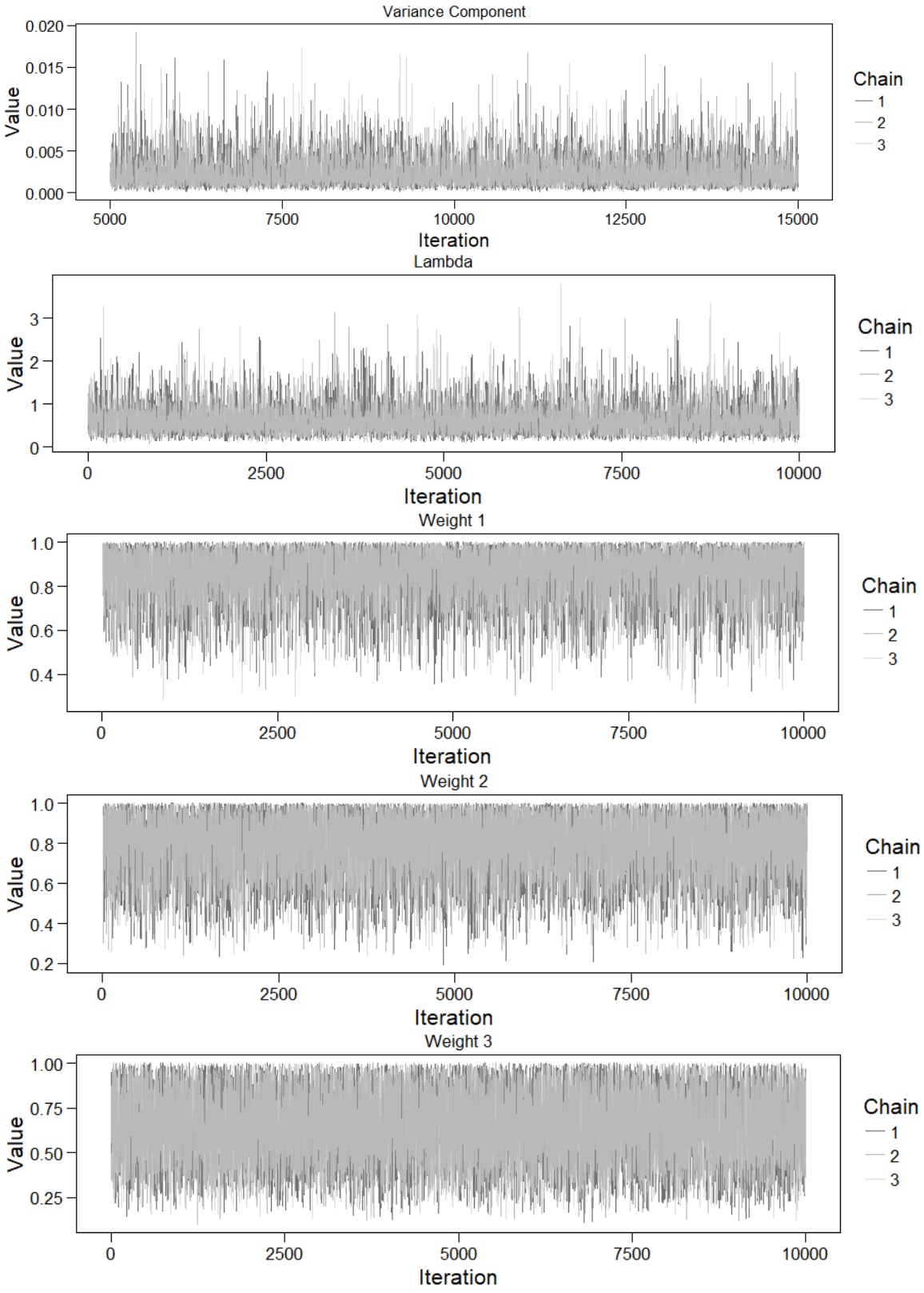
Trace plots are a crucial aspect of assessing convergence. They are essentially a line graph of the time series for each parameter, plotting the values of the parameter estimates by iteration. If there are multiple chains, the chains are often plotted on top of one another, differentiating by color, which also facilitates comparisons between the chains (Fernández-i-Marín, 2016). The trace plots contain only the post-burn-in iterations, so the chains should have "forgotten" their starting values and stabilized, and the trace plot should show white noise, a fuzzy blur. Trace plots displaying no evidence of non-convergence are often described as caterpillars; for an ideal trace plot, see Figure 31.

Figure 31. An example of the ideal trace plot.

Trace plots for all estimated parameters are presented in Figure 32.

If there is some sign of pattern or tendency in the time series of the chains, this indicates potential non-convergence; the objective is for a trace plot to appear random. Note that the trace plot for the mean resembles Figure 31 quite well. There are some spikes, but overall these spikes appear to be nothing more than random noise. In comparison, the trace plot for the variance component is bounded by the lower limit; the variance component cannot be negative, and therefore its parameter estimate cannot be lower than zero. The trace plot for λ is bounded by its lower limit of zero. For these cases, and in the subsequent trace plots of the weight estimates, notice that the upper or lower bounds of the parameter estimates influence the behavior of the chains. This is reasonable and not cause for concern; the trace plot for the variance component is often affected in such a way. Overall, the trace plots do not appear distinctly abnormal (for meta-analytic trace plots).




Variance Component
Value
0.020
0.015
0.010
0.005
0.000
5000
7500
10000
12500
15000
Iteration
Chain
1
2
3
Lambda
Value
3
2
1
0
0
2500
5000
7500
10000
Iteration
Chain
1
2
3
Weight 1
Value
1.0
0.8
0.6
0.4
0
2500
5000
7500
10000
Iteration
Chain
1
2
3
Weight 2
Value
1.0
0.8
0.6
0.4
0.2
0
2500
5000
7500
10000
Iteration
Chain
1
2
3
Weight 3
Value
1.00
0.75
0.50
0.25
0
2500
5000
7500
10000
Iteration
Chain
1
2
3

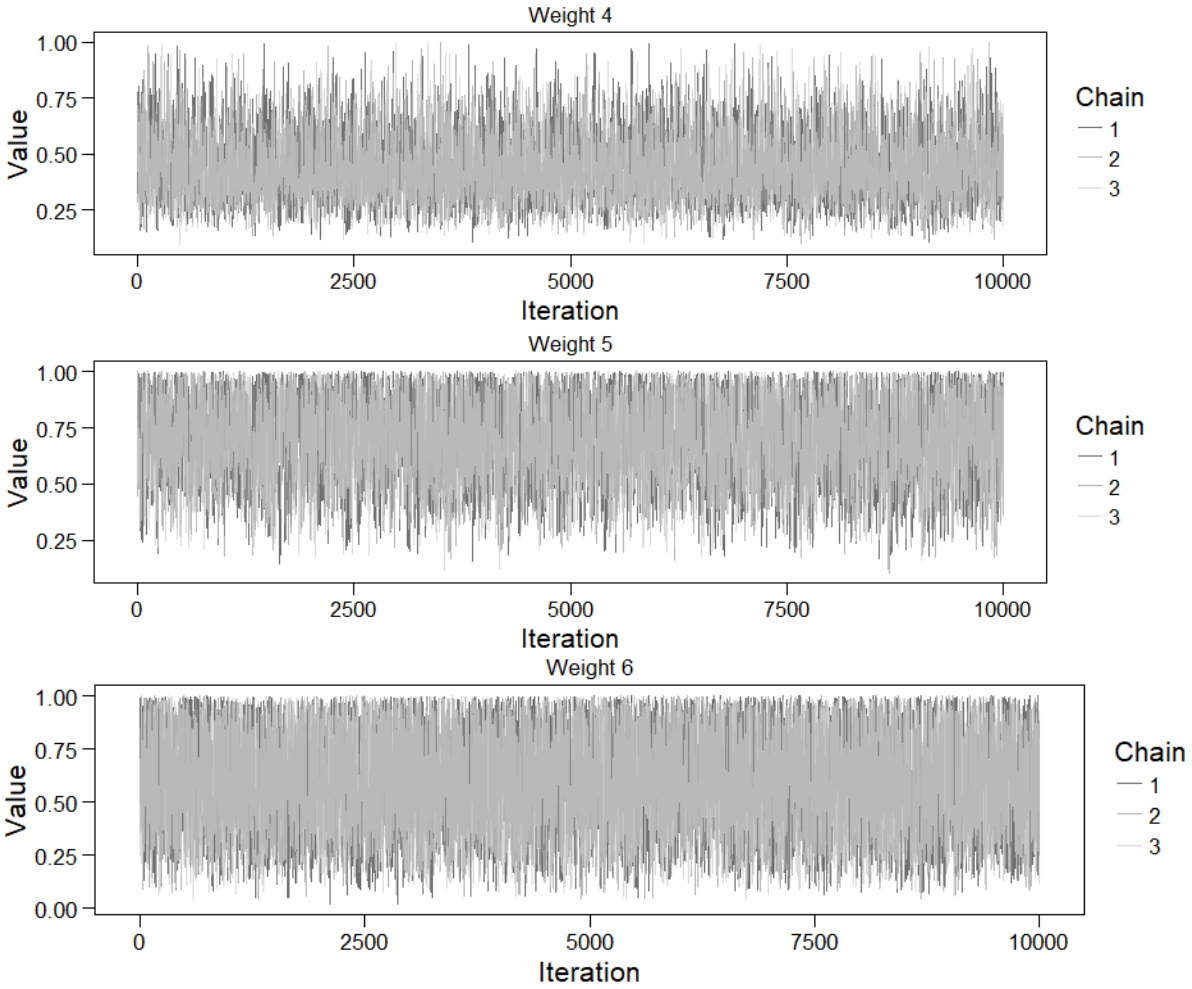
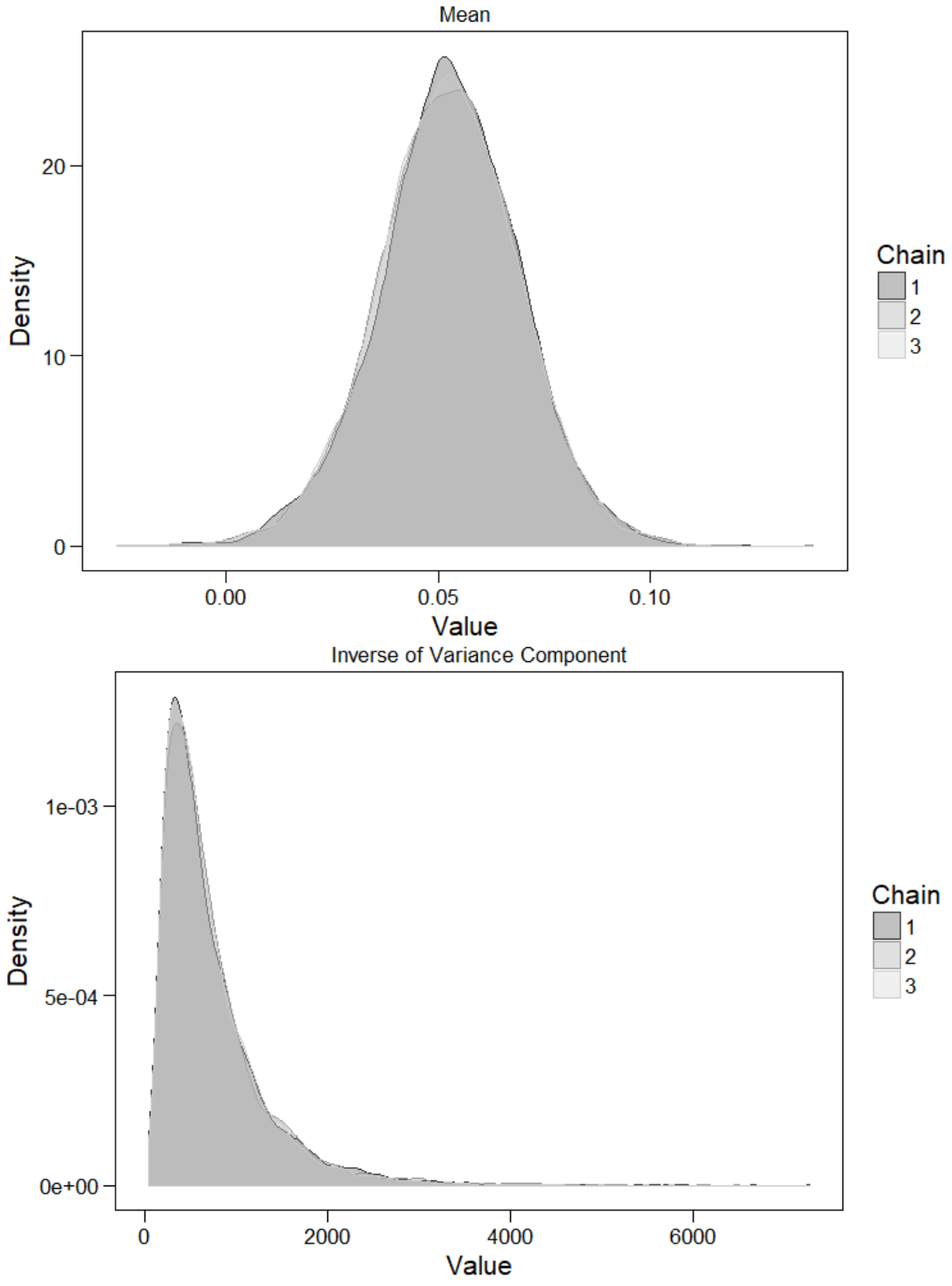


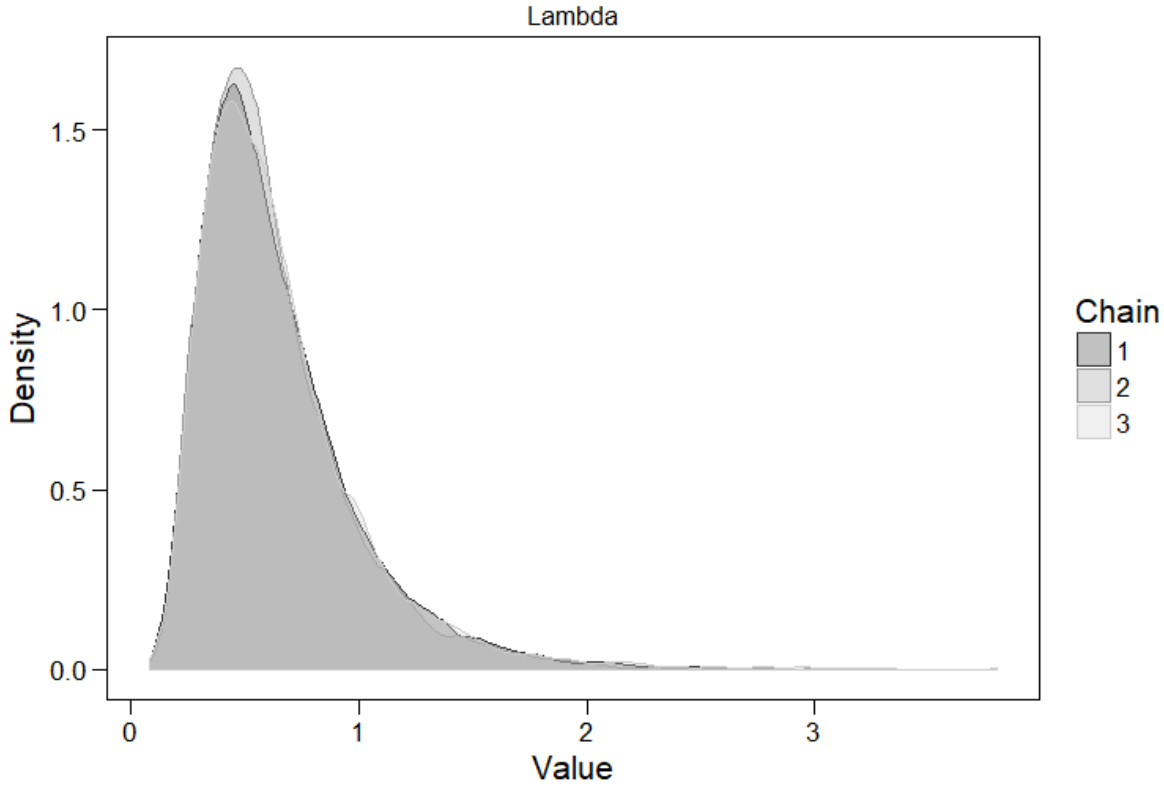
Figure 32. Trace plots for all parameters, Bem data.

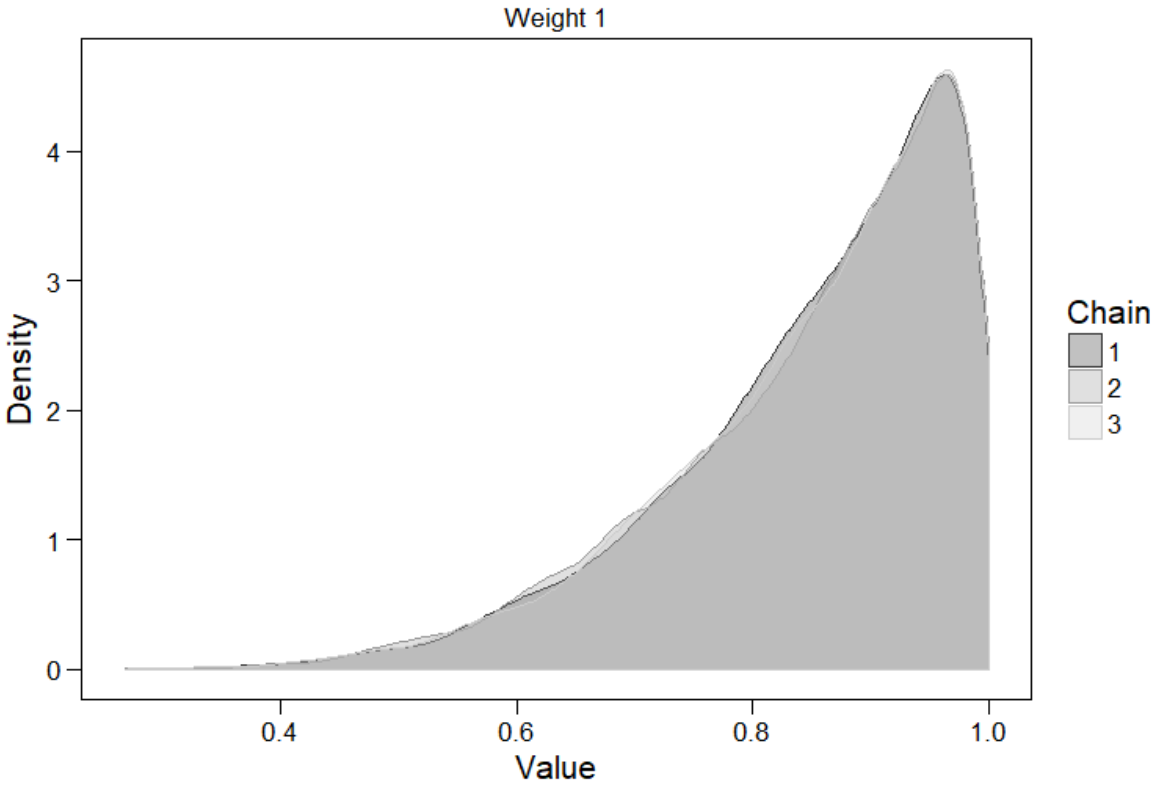
Now that we have calculated a range of convergence diagnostics and manually examined the trace plots for each of the parameter estimates, and because we have found no evidence of non-convergence, we can proceed to the posterior distributions of the parameters.

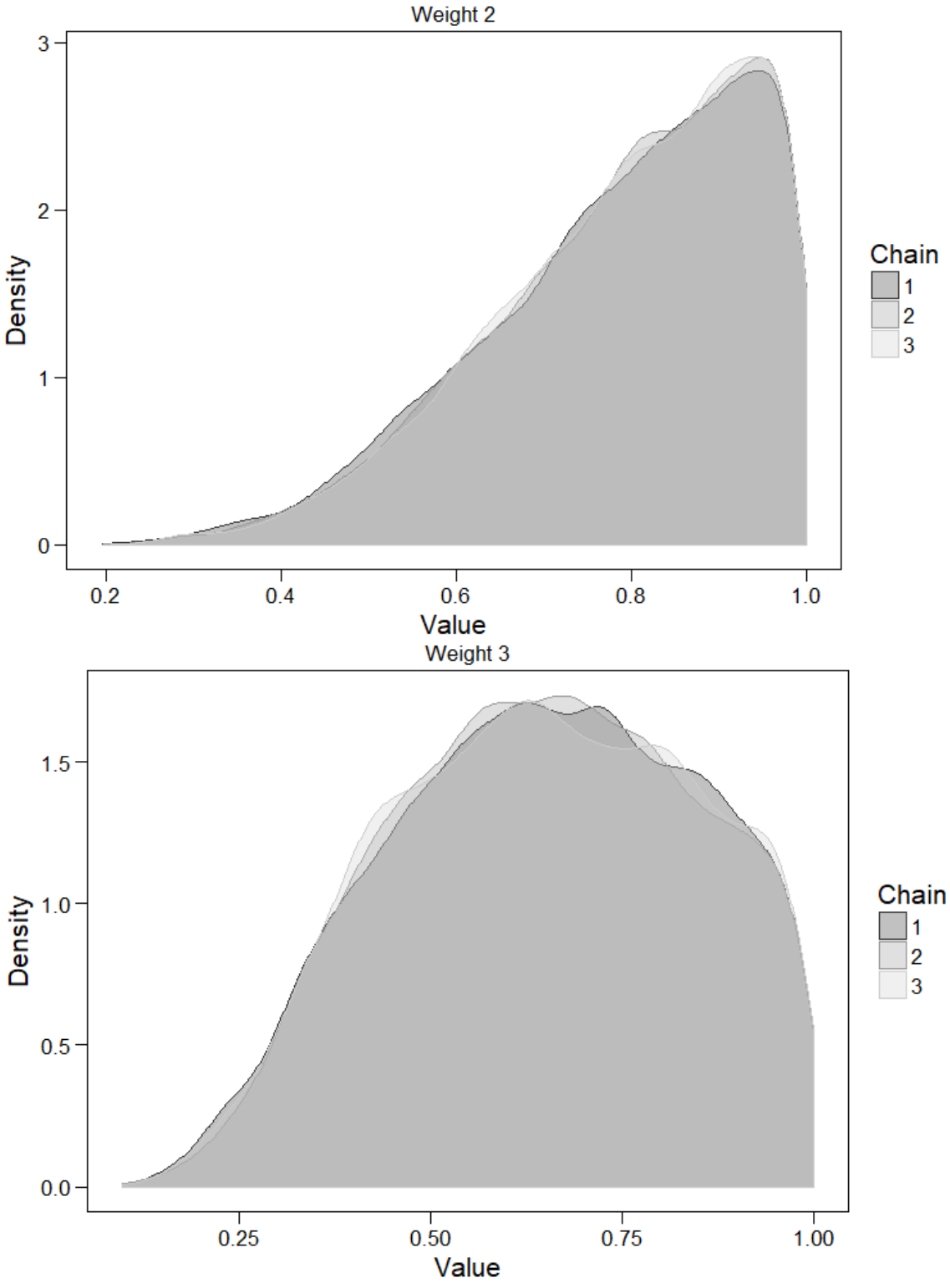
6.2.2 Model Results

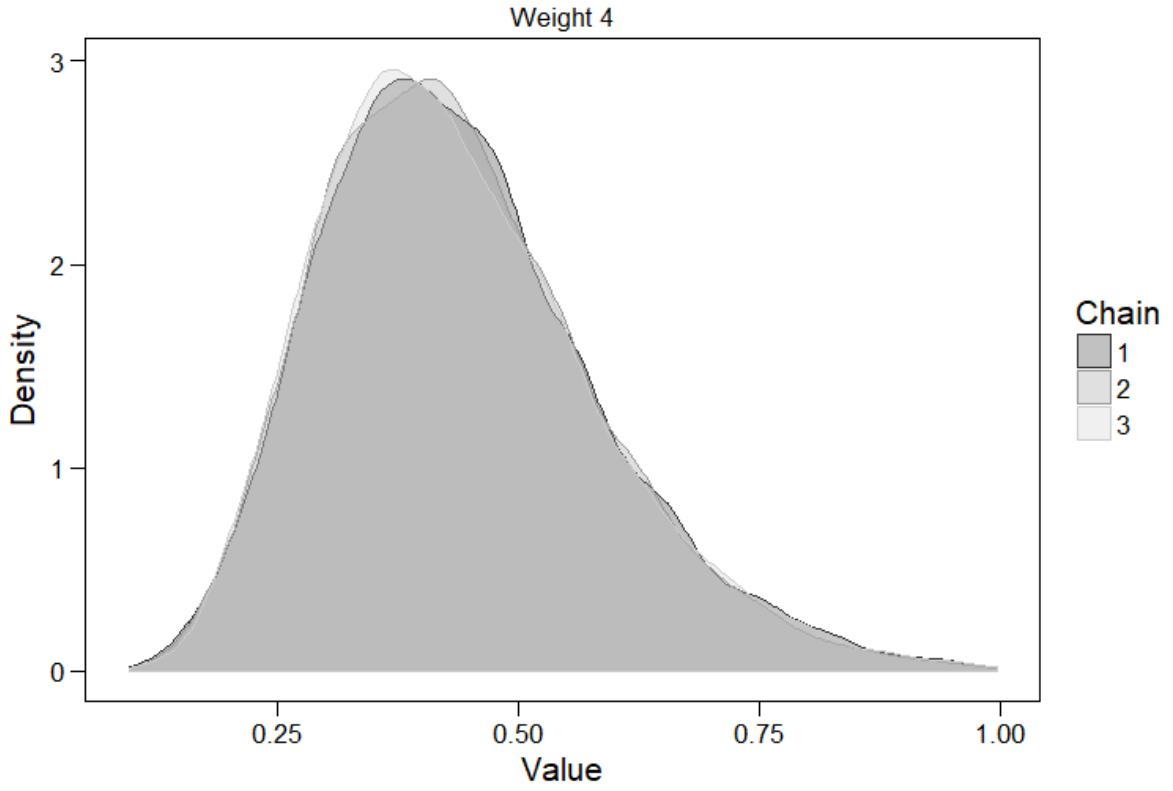
It is useful to examine a histogram, or a density plot of the posterior broken down by chain; this provides additional information about whether the chains have converged in the same place and allows for comparison across chains (Fernández-i-Marín, 2016). The following density plots represent each chain with a slightly different shade, denoted in the legend. The density plots for the posterior distributions of all parameters are presented in Figure 33. Table 6 presents the mean, median, and standard deviation of each posterior distribution. Note that, although thus far I have referred to the distribution of the *inverse* of the variance component, Table 6 presents the summary statistics transformed to present the variance component in its original metric for interpretation.

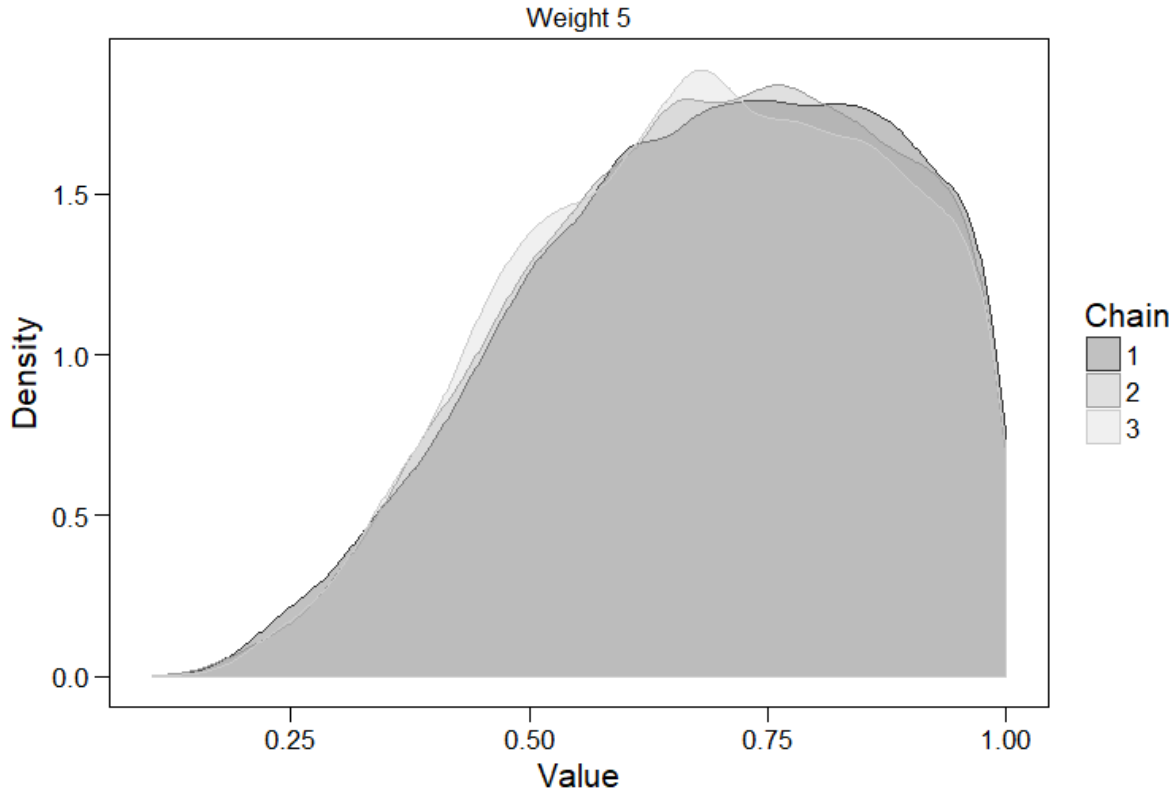

Mean
Density
20
10
0
0.00
0.05
0.10
Value
Chain
1
2
3
Inverse of Variance Component
Density
1e-03
5e-04
0e+00
0
2000
4000
6000
Value
Chain
1
2
3


Lambda
Density
1.5
1.0
0.5
0.0
0
1
2
3
Value
Chain
1
2
3


Weight 1
Density
4
3
2
1
0
0.4
0.6
0.8
1.0
Value
Chain
1
2
3


Weight 2
Density
Value
Chain
1
2
3
Weight 3
Density
Value
Chain
1
2
3


Weight 4
Density
3
2
1
0
0.25
0.50
0.75
1.00
Value
Chain
1
2
3


Weight 5
Density
1.5
1.0
0.5
0.0
0.25
0.50
0.75
1.00
Value
Chain
1
2
3

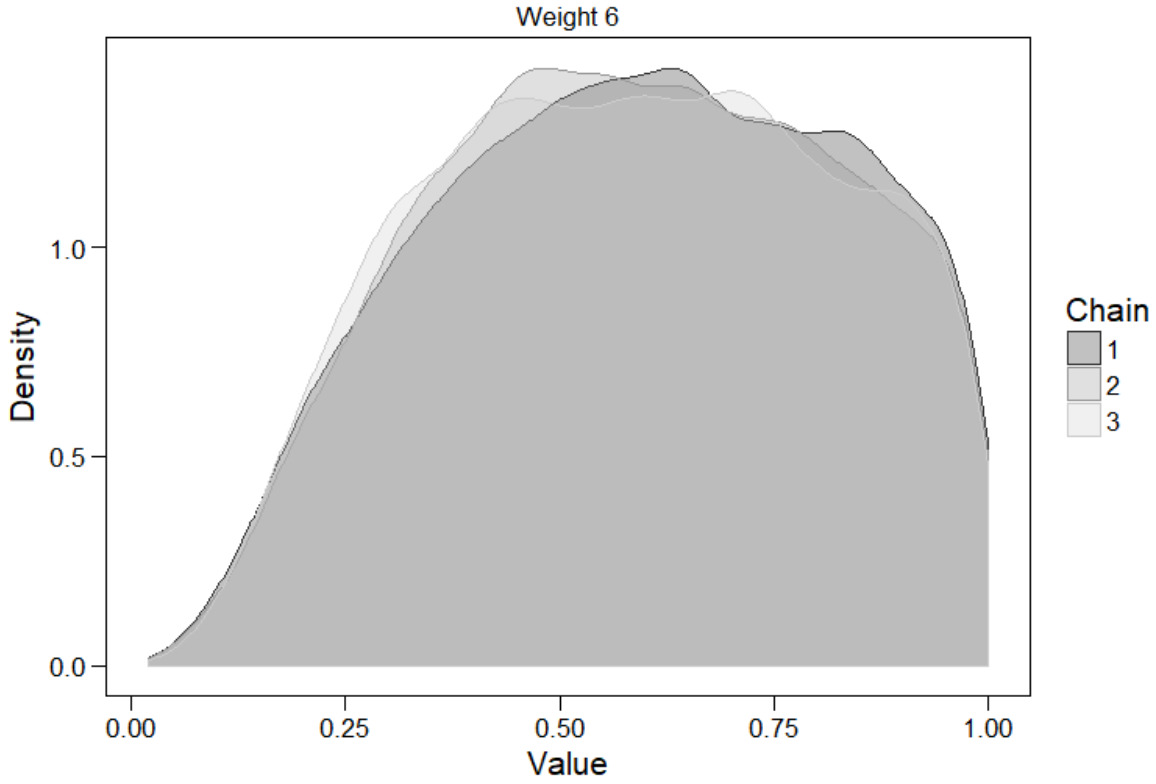


Figure 33. Posterior distributions of all parameters, Bem data.

Table 6. Posterior distribution of the Bem dataset, summary statistics.

Parameter	Mean	Standard Error	ML Estimates
Intercept	0.05259	0.01671	0.02283
Variance Component	0.00137	0.00174	N/A
Lambda (λ)	0.64140	0.34553	0.86589
Weight 1	0.85161	0.12046	1.00000
Weight 2	0.79033	0.15156	0.47020
Weight 3	0.64742	0.19612	0.24681
Weight 4	0.43765	0.14283	0.28793
Weight 5	0.68756	0.18621	0.26683
Weight 6	0.58967	0.22930	0.54729

Unlike maximum-likelihood estimation, which produces point estimates, Bayesian estimation yields entire distributions for each parameter, and so permits the researcher to directly estimate the standard errors of the parameters. Therefore, the Bayesian standard errors are included in Table 6.

We can refer to Table 3 to compare the Bayesian parameter estimates with the parameter estimates from the maximum-likelihood version of the lambda model. (I have also added those maximum-likelihood estimates to Table 6 for convenience.) One glaring difference is the estimate of the variance component; with maximum-likelihood estimation, we were unable to estimate a random-effects model at all and had to settle for a fixed-effect model. The Bayesian model does not reduce the estimate of the mean quite as far as the maximum-likelihood version; this is likely partially due to the estimation of the variance component and partially because the pattern of weights in the Bayesian version is not quite as drastic. Of course, remember that the weights are not directly comparable. For instance, to get an estimate of Weight 2 from the Bayesian model in the maximum-likelihood context, we would calculate its value relative to the value of Weight 1: 0.79033/0.85161 = 0.92804, a less severe weight.

It is also worth noting that some of the weights in the Bayesian context may be less extreme because information from the prior distribution can compensate for a lack of observed data in each interval. Weight 5 is an example of a case where the Bayesian estimate is larger. There are fewer observed effect sizes in that *p*-value interval, so the mean of the prior distribution (0.50, the mean of a uniform distribution from 0 to 1) may have more impact. This is not necessarily a flaw. It likely just means that, in the case of smaller datasets, the Bayesian version of the lambda model may err on the side of conservatism.

In the next section, I discuss the results of a simulation (proposed in Chapter 3) to explore the performance of the Bayesian lambda model.

6.3 Simulation Results

I conducted approximately 10,000 replications per cell. For each replication, I estimated the Bayesian lambda model with the priors described previously. I used three chains for each model, with 1,000 burn-in and 5,000 post burn-in iterations per chain, and a thinning interval of 1, meaning that there was no thinning and each iteration was retained. The results of the simulation are described below.

6.3.1 Convergence

This Bayesian simulation is very large, and all its corresponding data were stored. Given its collective size, the individual data are not particularly easy to manipulate. As a result, I assess model convergence using two diagnostic tests, and then survey trace plots from a few individual models.

Rather than assessing Gelman's potential scale reduction factor for each individual parameter in each of these models, I used what is known as the global potential scale reduction factor (GPSRF) to assess convergence of the entire model. The GPSRF automatically retains the maximum difference among all individual PSRFs for a model.

Across all cells, the average GPSRF was 1.016. The median of the distribution of GPSRF was 1.013, and the maximum GPSRF across all cells was 1.043, with a third quartile of 1.023. Guidelines indicate that PSRF below 1.10 suggest adequate convergence (Gelman & Rubin, 1992). Based on this rule of thumb, none of the cells display evidence of non-convergence. Of course, rules of thumb are not always reliable,

so it is useful to assess model convergence using other methods as well. With this in mind, I also used the Geweke (1992) diagnostic, as described previously.

Across all cells, the Geweke (1992) diagnostic test was non-significant for every single parameter. The average *p*-value of the diagnostic test per parameter ranged from a low of $p = 0.45$ to a high of $p = 0.49$, and no *p*-values were lower than 0.36. This also indicates that none of the cells displayed evidence of non-convergence.

Finally, as demonstrated, it is very useful to examine trace plots of the parameter estimates. The reader can likely imagine that doing this for every replication would be cumbersome. I do not present additional trace plots here. However, for each of several models that I selected and assessed, the trace plots did not differ much from those presented as a substantive example. With that, and with the results of the PSRF and the Geweke (1992) diagnostic, I conclude that the models, in general, do not bear cause for concern.

6.3.2 Mean Estimate

The results for the average estimate of the mean are presented much as they were in Chapter 4. The plots presented here compare the average estimate of four models – the Bayesian lambda model, the maximum likelihood version of the lambda model, an unadjusted meta-analytic model, and the original Vevea and Hedges (1995) weight-function model. A single horizontal line at 0.20 represents the population mean pre-selection, and the models are distinguished by different line types.

Again, I begin with cells where the selection mechanism matches the model. Figure 34 displays the results for cells where I^2 is 0%, and Figure 35 displays those for cells where I^2 is 75%.

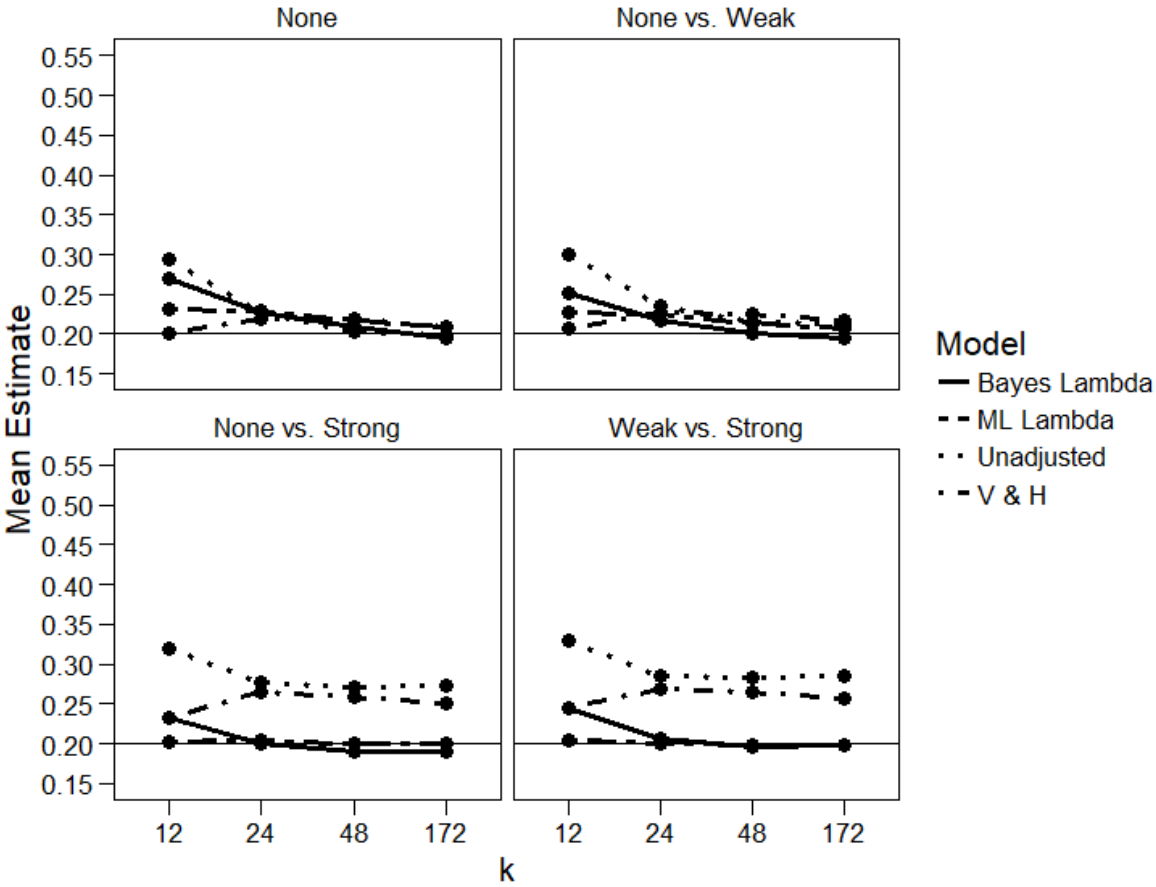


Figure 34. Estimates of the mean from cells with I^2 of 0%, bias generated with Method 1.

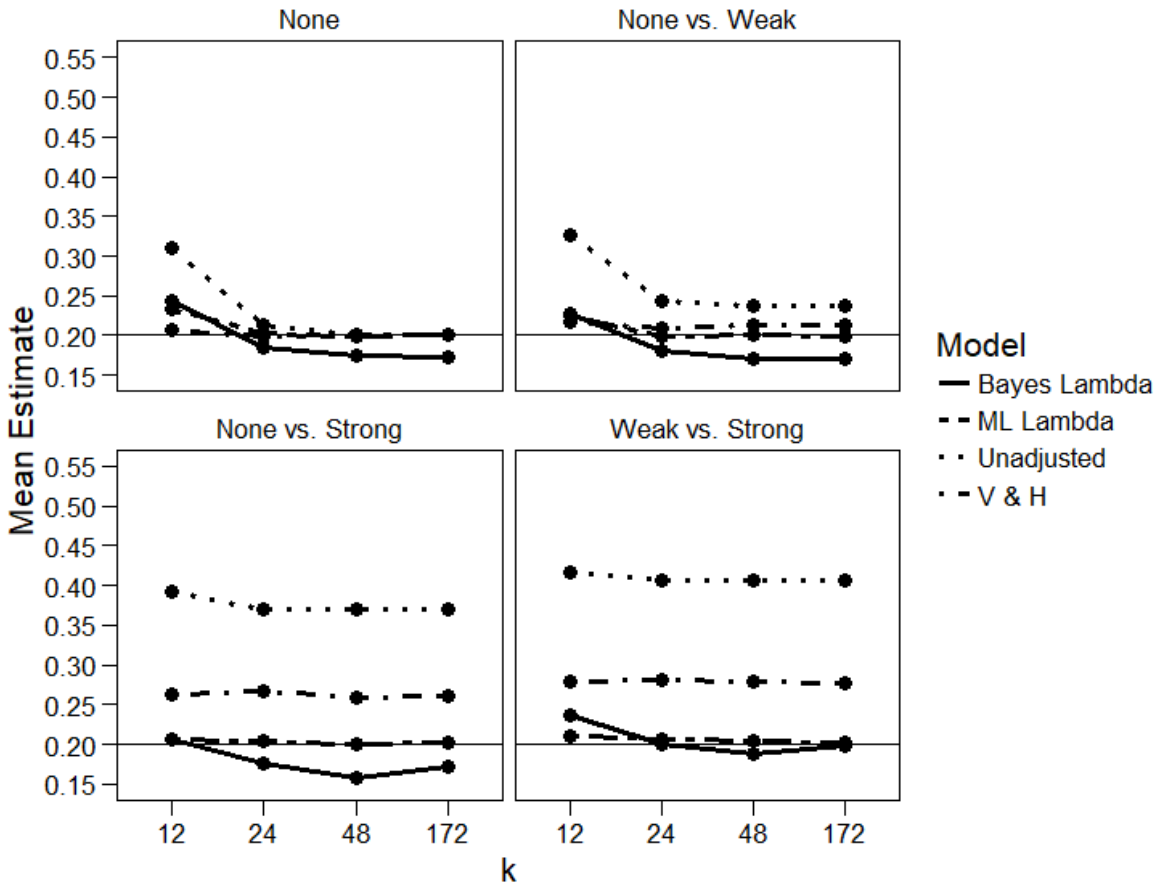


Figure 35. Estimates of the mean from cells with I^2 of 75%, bias generated with Method 1.

When I^2 is 0%, the Bayesian version of the lambda model performs almost identically to the maximum likelihood version, except for a few cases where the Bayesian version slightly underestimates the mean. When I^2 is 75%, the maximum likelihood estimate is more accurate in most cases, while the Bayesian model is at least a slight underestimate.

This may be the case for a few reasons. Bayesian estimation differs from maximum likelihood estimation in some fundamental ways. It is possible that these cases of underestimation are due to the influence of the prior distribution(s). It is also possible that the results would differ if the initial values were changed, or if the models were run for more iterations.

Figure 36 and Figure 37, respectively, display the results of cells where I^2 is 0% and selection bias was generated as an exponential function of *p*-value (Method 2, as described in Chapter 3).

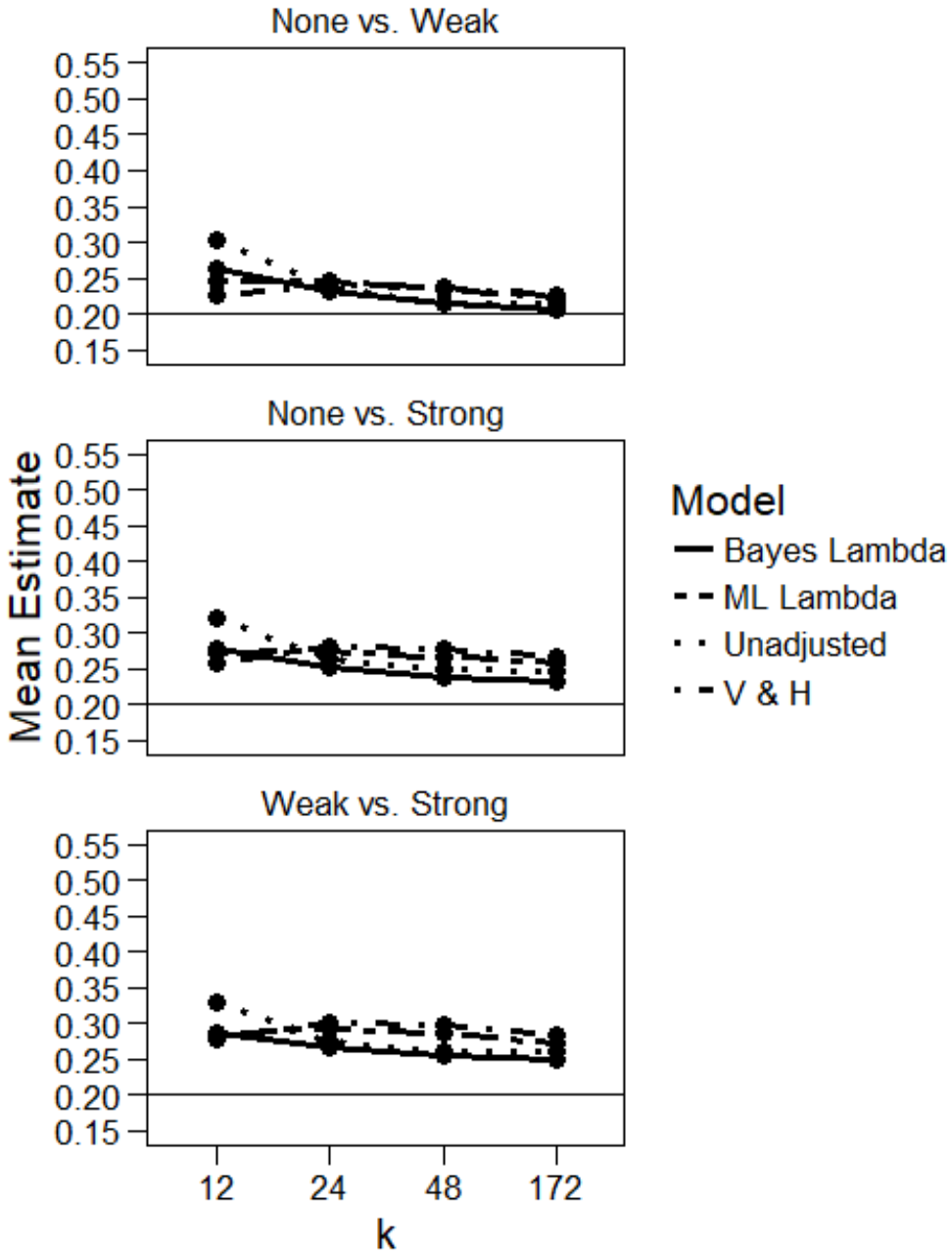


Figure 36. Estimates of the mean from cells with I^2 of 0%, bias generated with Method 2.

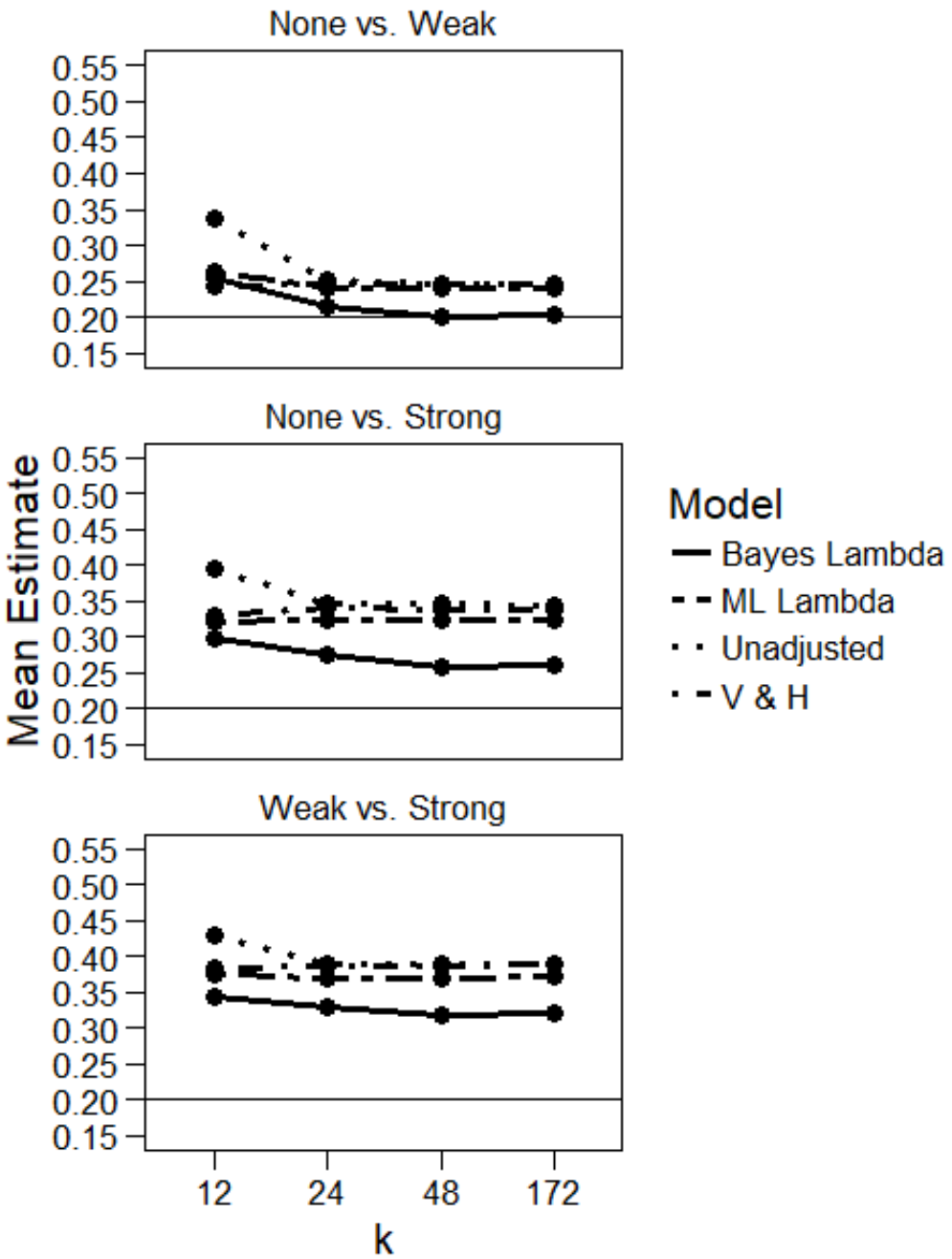


Figure 37. Estimates of the mean from cells with I^2 of 75%, bias generated with Method 2.

These figures reveal something unexpected and encouraging. Across cells, when bias is *not* generated according to model specifications, the Bayesian version of the lambda model produces a more accurate estimate of the mean. Although it sometimes underestimates the mean, it performs better than the maximum likelihood version when model assumptions are violated – sometimes much better.

This pattern is continued in Figure 38, Figure 39, Figure 40, and Figure 41, which display the results of cells with 0% and 75% I^2 where bias was generated according to Method 3 and Method 4.

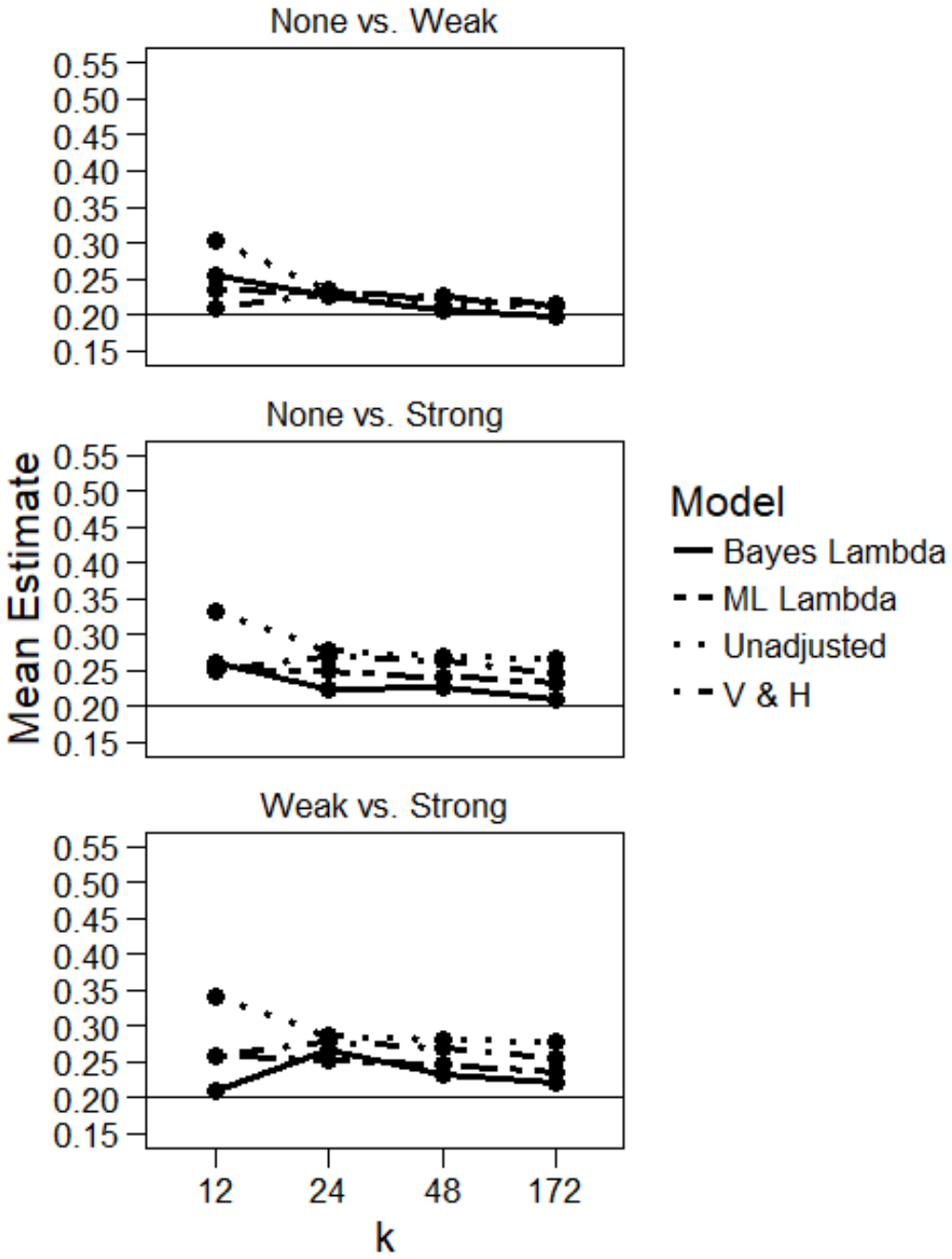


Figure 38. Estimates of the mean from cells with I^2 of 0%, bias generated with Method 3.

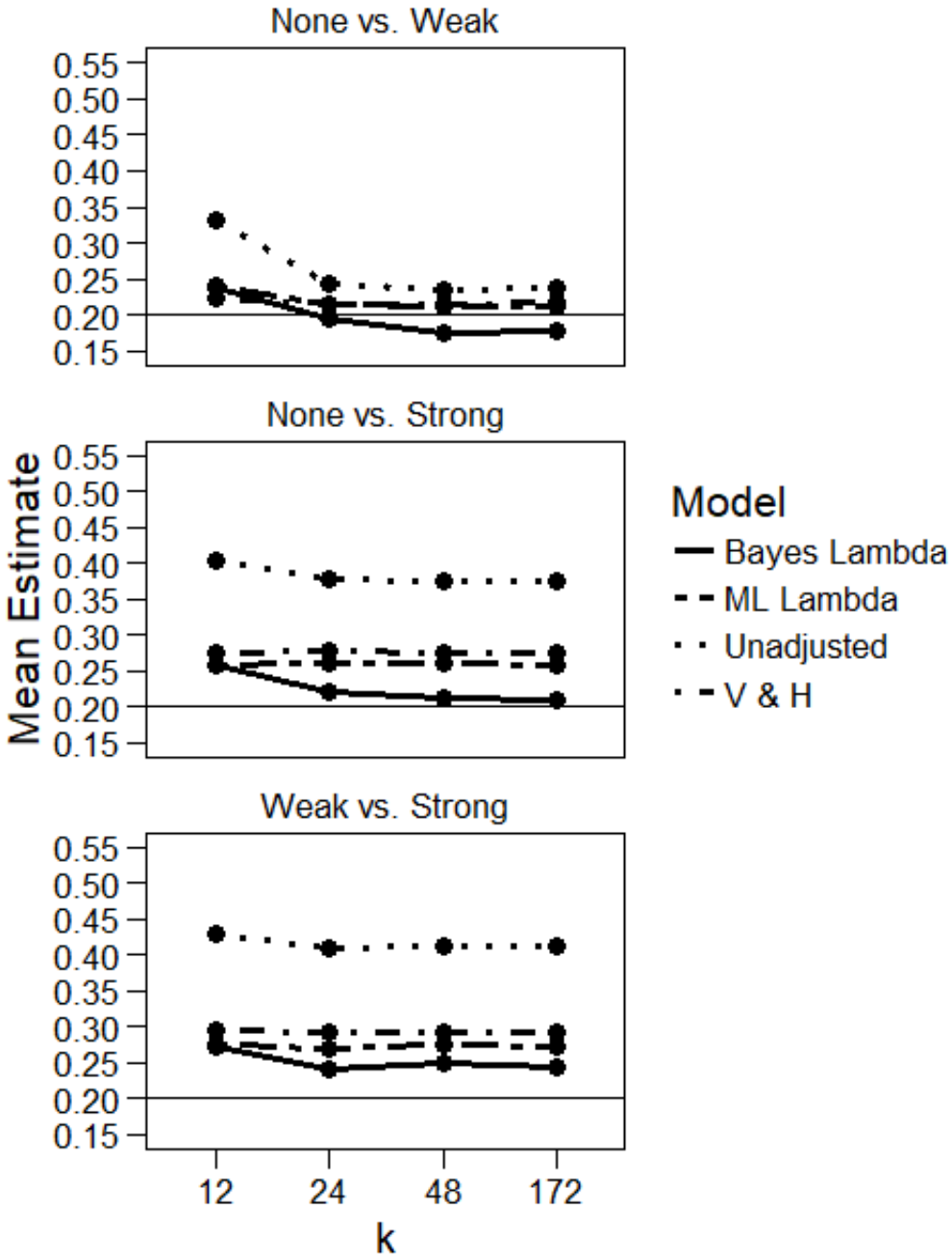


Figure 39. Estimates of the mean from cells with I^2 of 75%, bias generated with Method 3.

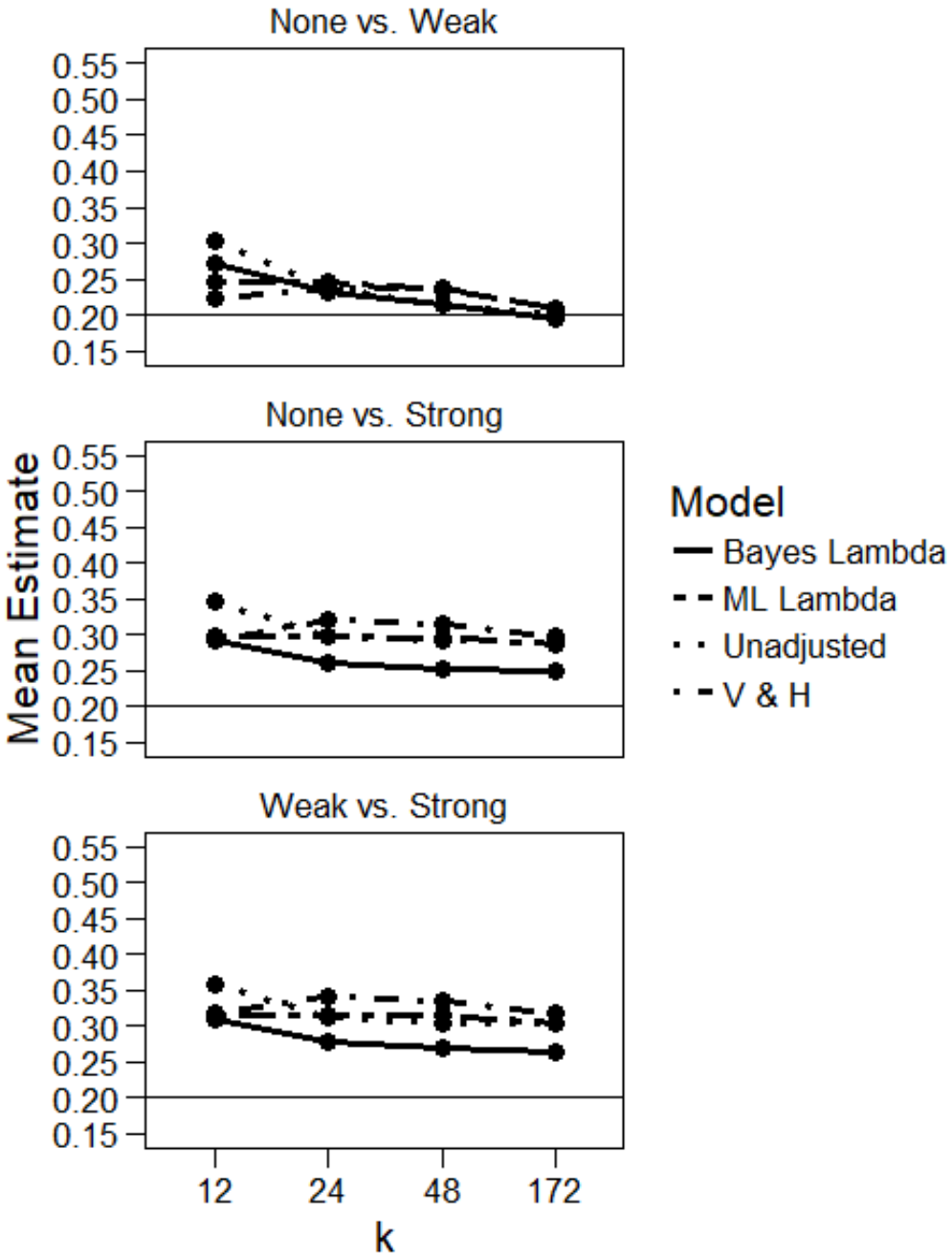


Figure 40. Estimates of the mean from cells with I^2 of 0%, bias generated with Method 4.

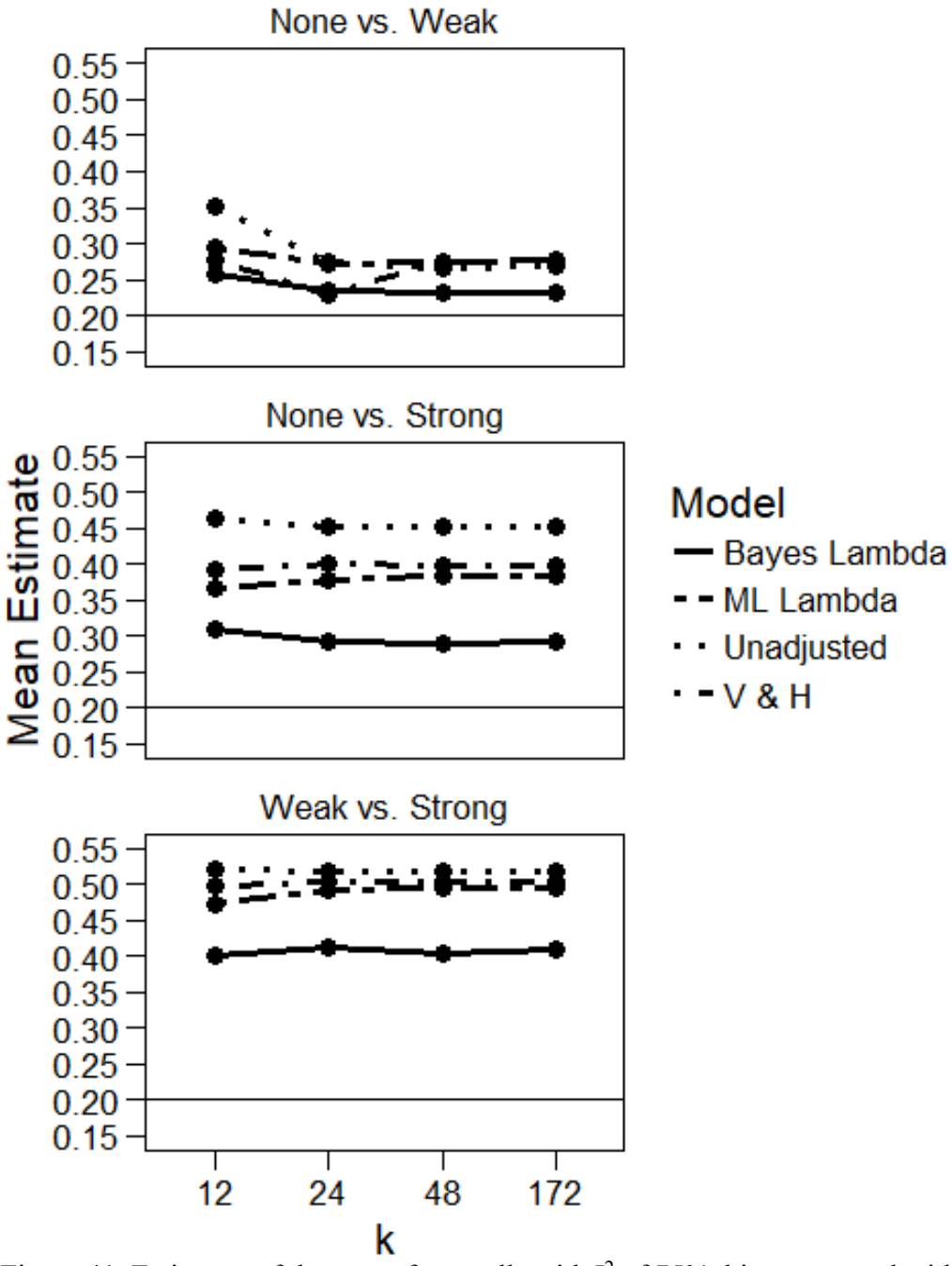


Figure 41. Estimates of the mean from cells with I^2 of 75%, bias generated with Method 4.

Across cells, the Bayesian model continues to perform better than its maximum likelihood counterpart when bias is not generated according to the model assumptions. These results are very encouraging.

The next section assesses the Bayesian estimate of the lambda parameter.

6.3.3 Lambda Estimate

Figure 42 displays the results of cells where I^2 is 0%, and Figure 43 the cells where I^2 is 75%. Again, the gray lines represent approximately ideal values of lambda per bias pattern. The line types in these plots represent levels of bias pattern.

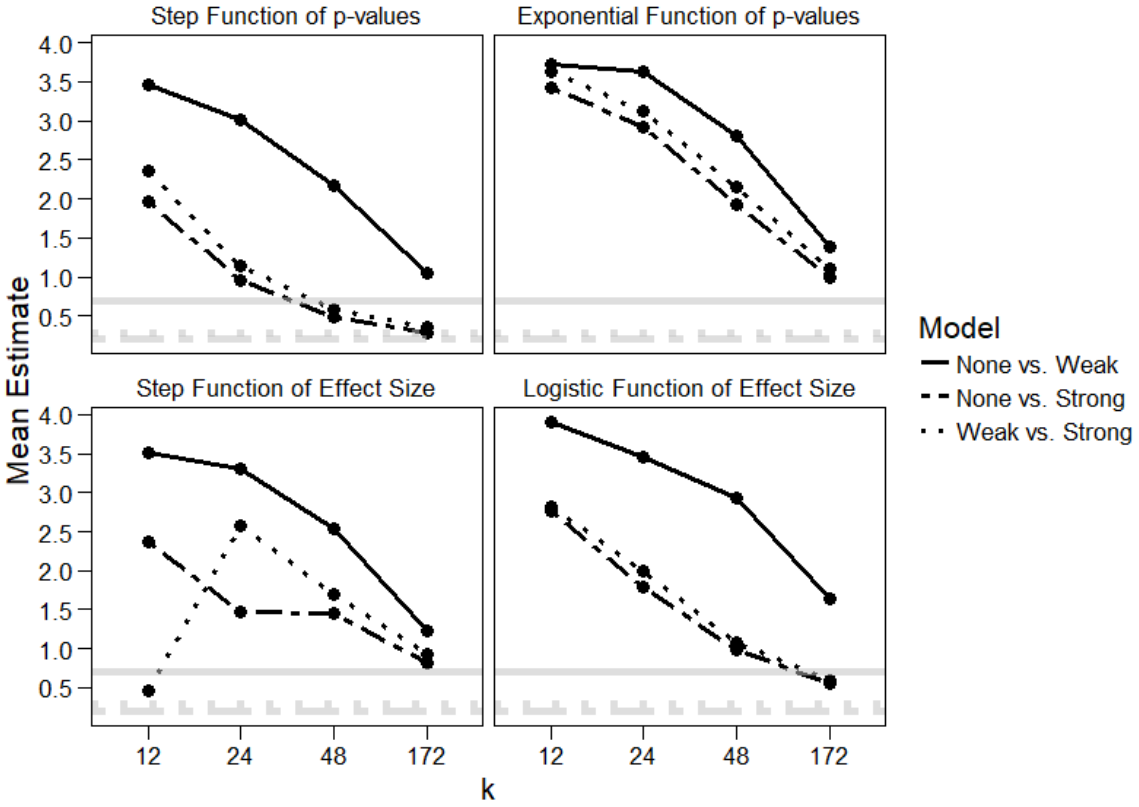


Figure 42. Estimates of lambda across methods of bias generation and bias pattern from cells with I^2 of 0%.

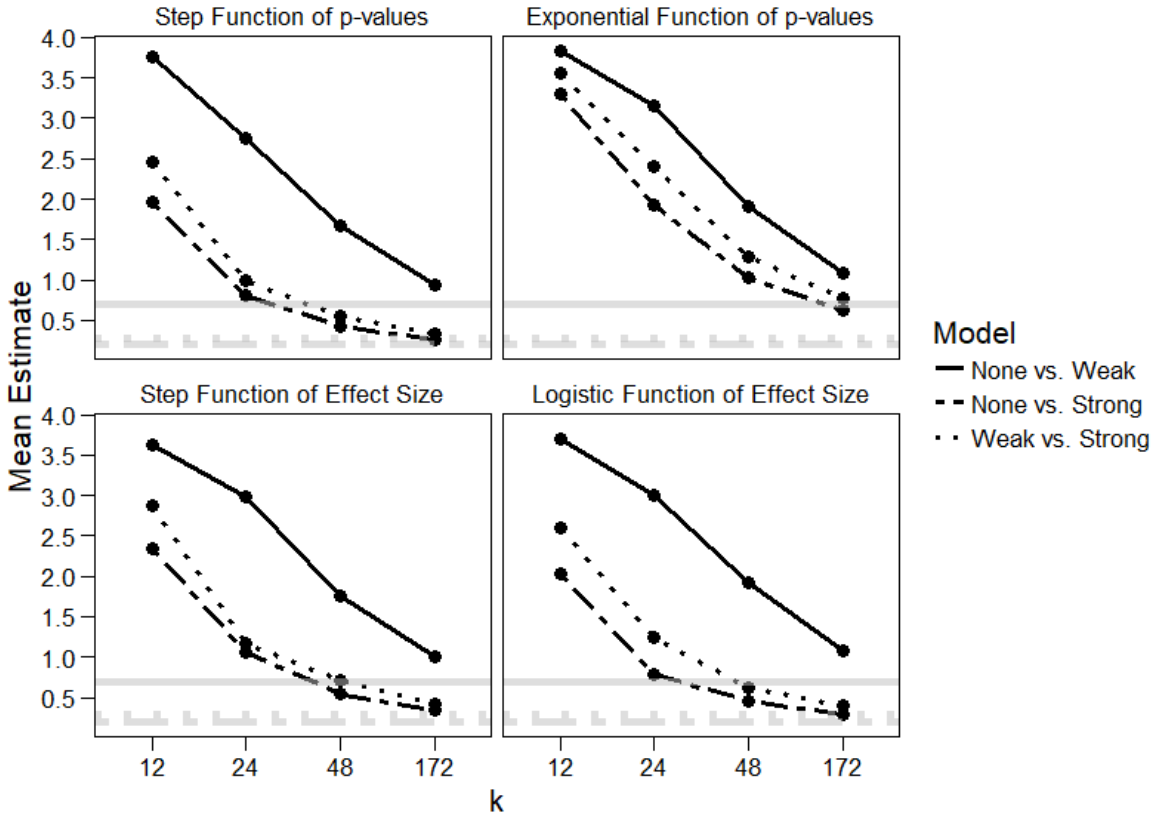


Figure 43. Estimates of lambda across methods of bias generation and bias pattern from cells with I^2 of 75%.

In terms of the lambda parameter, the Bayesian model performs very similarly to the maximum likelihood model. By the time the number of studies has reached 172, the estimate is fairly accurate. When *k* is small, around 12, the estimate of lambda is worse, which is to be expected, especially since the lambda parameter is more difficult to estimate accurately than the mean effect size.

6.4 Conclusions

The Bayesian version of the lambda model is surprisingly robust to violations of model assumptions; in fact, it is more robust in most cases than the maximum likelihood version. If meta-analysts bear in mind that the Bayesian version of the lambda model can sometimes be overly conservative, the Bayesian version can be very useful. In fact, if there is a reason to believe that model assumptions have been violated, the Bayesian version may be preferable.

Future work should involve exploring the Bayesian model further. Changing the specified priors, the initial values, the number of chains, and the number of burn-in or post burn-in iterations may affect the performance of the model and understanding such changes in performance would be informative.

Chapter 7: *weightr*: Software for Model Implementation

Software design, although often overlooked by researchers and statisticians, is arguably the most important area of model development. The reason is simple – if no one can use the model, there is no reason for the model to exist. A statistician might develop the "perfect" model – one with no bias, absolute accuracy, and capable of explaining all variance – but such a model is irrelevant if it cannot be used; it is a theory, not a tool.

It is possible to argue that empirical researchers should not need software to implement models, or that anyone interested in using a model should be capable of writing the code to implement it themselves. In the case of calculating a mean or a *t*-test, that is true. However, as models become more complicated, so does their implementation. An empirical researcher may be very interested in using structural equation modeling to assess a phenomenon but sitting that researcher down before a blank *R* terminal is unlikely to yield much success. It is implausible to expect an empirical user to be willing to dedicate as much time and effort to implementing a model as the models' creators. Maintaining such an expectation will result in nothing but disappointment and may prevent the model from ever being used at all. Therefore, it is in statisticians' best interests to ensure that the models they develop are both widely available and (relatively) easy to estimate.

Sometimes, as with selection modeling and publication bias, researchers may agree that a certain class of models performs best in a given situation, but the models are difficult to estimate or no software to do so is available. As a result, the area of research may stagnate, while empirical researchers use more convenient (but less effective) tools (e.g., the failsafe-*N*). For example, in publication bias assessment, many articles will note that selection models perform best, but very few meta-analysts employ selection models (Ferguson & Brannick, 2012); instead, the failsafe *N* is most commonly used, despite its long list of well-documented flaws. The Vevea and Hedges (1995) model was first published over 20 years ago but has received only 145 citations since then, a small number considering the model's performance and capabilities. The primary reason for the model's limited use is almost certainly the fact that it was, up until about two years ago, difficult to access. In contrast, the Vevea and Woods (2005) model was published with usable code and, despite being 10 years more recent, has been cited 188 times.

Some statistical software, like SPSS, SAS, and Comprehensive Meta-Analysis (to name a few), is proprietary, meaning that the author of said software can restrict or prevent any modifications of it. Fortunately, though, the research world has recently begun to turn away from proprietary programs and toward free, open-source software like *R* (R Core Team, 2017). Rather than struggling to write and submit a macro for a brand-new model that can fit into pre-existing software, statisticians can use *R* to create their own software in the form of *R* packages containing functions and even point-and-click applications. This software is then available to any interested parties, who can easily implement it themselves by accessing *R* and loading the required package. As of this writing, there are 12,726 packages available for download and installation on the Comprehensive *R* Archive Network (CRAN), the most common source of package distribution, and innumerable packages available on GitHub, a popular git network for package development. There are even conferences held focusing solely on *R* development and package usage, like useR! (a forum for the *R* community), rstudio::conf (about all

things *R* and RStudio), the Shiny Developer Conference (for developers of *R* Shiny applications), and more. Many well-known companies also use *R*, including Facebook, Google, Twitter, the New York Times, Microsoft, Zillow, and the Food and Drug Administration (FDA).

For these reasons and many more, *R* is an ideal host for today's budding programmer. I have used *R* to implement not only the lambda model and its variations presented here but also the Vevea and Hedges (1995) and the Vevea and Woods (2005) model. I developed a package dubbed *weightr* (for "weight-function models estimated in *R*," and pronounced "waiter"). This chapter is styled as a "package vignette" – the term for a document released in conjunction with an *R* package, intended for users as a tutorial or guide. During the vignette, any text meant to be entered at an R terminal is presented in Courier New font to distinguish it as a command.

Although a package vignette typically begins with an introduction fleshing out the concepts behind the software, I omit that section in lieu of the previous dissertation chapters, and begin.

7.1 The *weightr* Package

My *R* package, *weightr*, provides a single function that can estimate all the models described above.[6] *weightr* is available via the Comprehensive R Archive Network (CRAN), at *https://cran.r-project.org/package=weightr*, and can be installed directly through *R* by the commands install.packages("weightr") and library("weightr"), assuming that the user is connected to the Internet. The current version of *weightr* is 1.1.2, and it was last updated on April 4th, 2017. If CRAN detects that the version of *weightr* on a user's machine is out of date, it can either notify the user or update the package automatically. To any users who encounter "bugs," or problems, while working with *weightr*, I invite feedback and communication. My contact information as package maintainer is provided on CRAN.

7.1.1 Specifying Data

Before beginning to work with *weightr*, the user should note that *weightr* is not designed to handle all aspects of conducting meta-analyses. The package cannot, for instance, generate a forest plot or a cumulative meta-analysis. It does not possess an effect-size calculator. This is for a few reasons; first, many packages already exist that are more than capable of handling these aspects, including *metafor* (Viechtbauer, 2010). Second, that is not the purpose of *weightr*. *weightr* exists to do one primary task – that is, to estimate the Vevea and Hedges (1995) class of models – and to do it well.

Meta-analysts should not prepare to use *weightr* until they are equipped with a set of effect sizes and their corresponding sampling variances (as well as any moderator variables of interest). For meta-analysts in need of an effect-size calculator, *metafor* contains the function escalc(), and effect sizes can be extracted from said function with

[6] Again, note that the lambda model variants will not be publicly available until the models are formally published.

relative ease. For this tutorial, however, it is easiest to work with the two datasets contained within *weightr*, as they are automatically installed along with the package.

The first dataset, dat.bangertdrowns2004, is the smaller of the two, with *k* of 48. I have sourced the data from *metafor* (Viechtbauer, 2010), where it was extracted from a meta-analysis on the effects of school-based writing-to-learn interventions on academic achievement (Bangert-Drowns, Hurley, & Wilkinson, 2004). It contains some moderator variables. The second dataset, dat.gatb, is much larger; it consists of the results of 755 studies assessing the General Aptitude Test Battery (GATB)'s predictive validity of job performance. Although the actual GATB consists of several scales, this dataset only assesses the General Ability subscale. Interested users may learn more about the substantive aspects of these datasets through their CRAN documentation and citations.

We begin with the GATB data. Users can view the beginning of the data frame by entering the command head(dat.gatb); doing so will result in a screen like Figure 44.

```
> head(dat.gatb)
      z           v
1 0.321 0.02857143
2 0.192 0.03571429
3 0.266 0.03448276
4 0.255 0.04166667
5 0.288 0.05000000
6 0.161 0.02272727
```

Figure 44. The beginning of the GATB dataset in weightr.

The first column, z, contains the Fisher's *z*-transformed correlation coefficients; the second column, v, contains their corresponding sampling variances, calculated as 1/(N – 3). This dataset contains a column of effect sizes and a corresponding column of variances, so it is ready for use with *weightr*.

We can view the beginning of the Bangert-Drowns data by entering the command head(dat.bangertdrowns2004), which yields Figure 45:

```
> head(dat.bangertdrowns2004)
  id   author year grade length minutes wic feedback info pers imag meta       subject  ni     yi    vi
1  1 Ashworth 1992     4     15      NA   1        1    1    1    0    1       Nursing  60   0.65 0.070
2  2    Ayers 1993     2     10      NA   1       NA    1    1    1    0 Earth Science  34  -0.75 0.126
3  3   Baisch 1990     2      2      NA   1        0    1    1    0    1          Math  95  -0.21 0.042
4  4    Baker 1994     4      9      10   1        1    1    0    0    0       Algebra 209  -0.04 0.019
5  5   Bauman 1992     1     14      10   1        1    1    1    0    1          Math 182   0.23 0.022
6  6   Becker 1996     4      1      20   1        0    0    1    0    0    Literature 462   0.03 0.009
```

Figure 45. The beginning of the Bangert-Drowns dataset in weightr.

This dataset contains more columns, most of which are moderator variables. The last two columns, yi and vi, contain the effect sizes and their corresponding sampling variances; therefore, this dataset, too, is ready for use with *weightr*.

7.1.2 Visualizing Data

It is inadvisable to begin an investigation of publication bias without first assessing the data visually, and a funnel plot is an excellent way of doing so. Conveniently, the function shiny_weightr() launches a local point-and-click interface that can produce both a funnel plot and a density plot. The user need only enter the command shiny_weightr() to launch the application. This interface is also available online, for those who have not installed *R* on a local machine, at *https://vevealab.shinyapps.io/WeightFunctionModel/*.

First, we will work with the GATB data. To read data into the Shiny application, users must have the dataset saved as a file on their computer. The application accepts several file formats and structures and allows users to specify whether their file contains a header and how it is separated. I export dat.gatb from *R* as a text file with the command `write.table(dat.gatb, "gatb.txt", sep="\t")`. Clicking on the Browse… button and navigating to the location of the data file yields Figure 46.

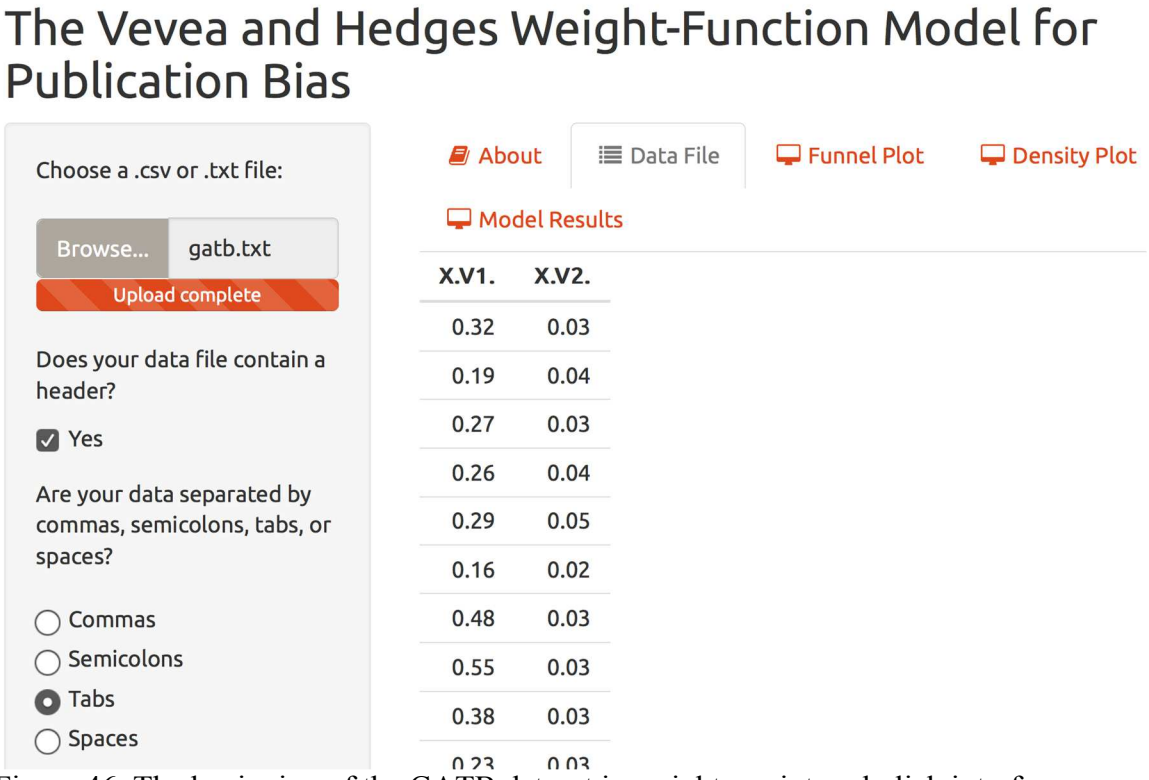


Figure 46. The beginning of the GATB dataset in weightr, point-and-click interface.

Note that I have selected some options in the menu on the left – namely, that the file contains a header and that the data are separated by tabs. Changing these options will change the display of the data on the right. Users need only select the correct options

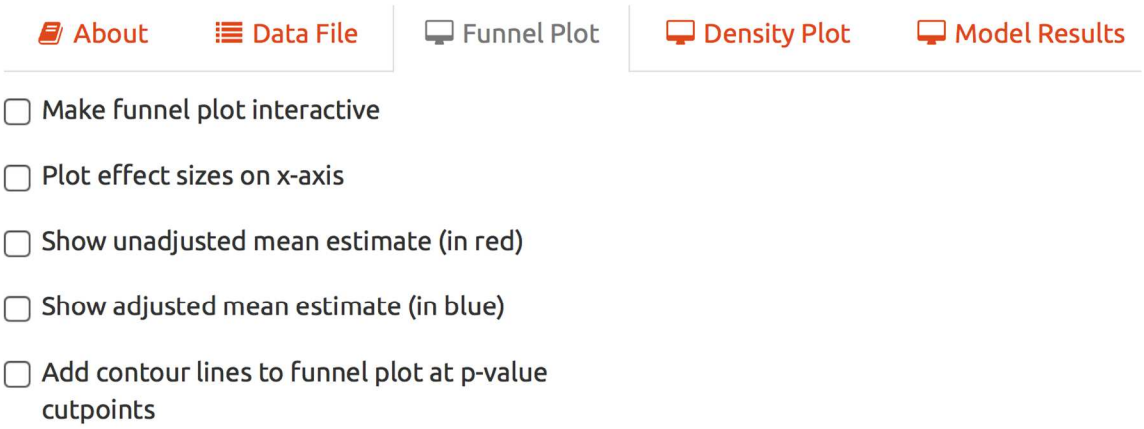
corresponding to their data file, or the options which result in the data displaying correctly. Here, the column X.V1. contains the Fisher's *z*-transformed correlation coefficients, and the column X.V2. their corresponding sampling variances.

I have also exported the Bangert-Drowns dataset, `write.table(dat.bangertdrowns2004, "bangertdrowns2004.txt", sep="\t")`, and I can upload this data as well. I do so here for the purposes of demonstration. However, take note that the Shiny application can accommodate only one dataset at once; in practice, switching back and forth is not likely to be practical. See Figure 47 for the beginning of the Bangert-Drowns dataset in Shiny.

X.id.	X.author.	X.year.	X.grade.	X.length.	X.minutes.	X.wic.	X.feedback.	X.info.	X.pers.	X.imag.	X.meta.	X.subject.	X.ni.	X.yi.	X.vi.
1	"Ashworth"	1992	4	15	NA	1	1	1	1	0	1	"Nursing"	60	0.65	0.07
2	"Ayers"	1993	2	10	NA	1	NA	1	1	1	0	"Earth Science"	34	-0.75	0.13
3	"Baisch"	1990	2	2	NA	1	0	1	1	0	1	"Math"	95	-0.21	0.04
4	"Baker"	1994	4	9	10	1	1	1	0	0	0	"Algebra"	209	-0.04	0.02
5	"Bauman"	1992	1	14	10	1	1	1	1	0	1	"Math"	182	0.23	0.02
6	"Becker"	1996	4	1	20	1	0	0	1	0	0	"Literature"	462	0.03	0.01
7	"Bell & Bell"	1985	3	4	NA	1	1	1	1	0	1	"Math"	38	0.26	0.11
8	"Brodney"	1994	1	15	NA	1	1	1	1	0	1	"Math"	542	0.06	0.01
9	"Burton"	1986	4	4	NA	0	1	1	0	0	0	"Math"	99	0.06	0.04
10	"Davis, BH"	1990	1	9	10	1	0	1	1	0	0	"Social Studies"	77	0.12	0.05
11	"Davis, JJ"	1996	4	15	NA	0	1	1	0	0	0	"Statistics"	40	0.77	0.11
12	"Day"	1994	4	15	NA	0	1	1	1	0	1	"Sociology"	190	0.00	0.02

Figure 47. The beginning of the Bangert-Drowns dataset in weightr, point-and-click interface.

Once the user has uploaded a data file and selected the columns of effect sizes and variances using the menu on the left, clicking the tab labeled Funnel Plot yields Figure 48 (for the GATB data) or Figure 49 (for the Bangert-Drowns data).



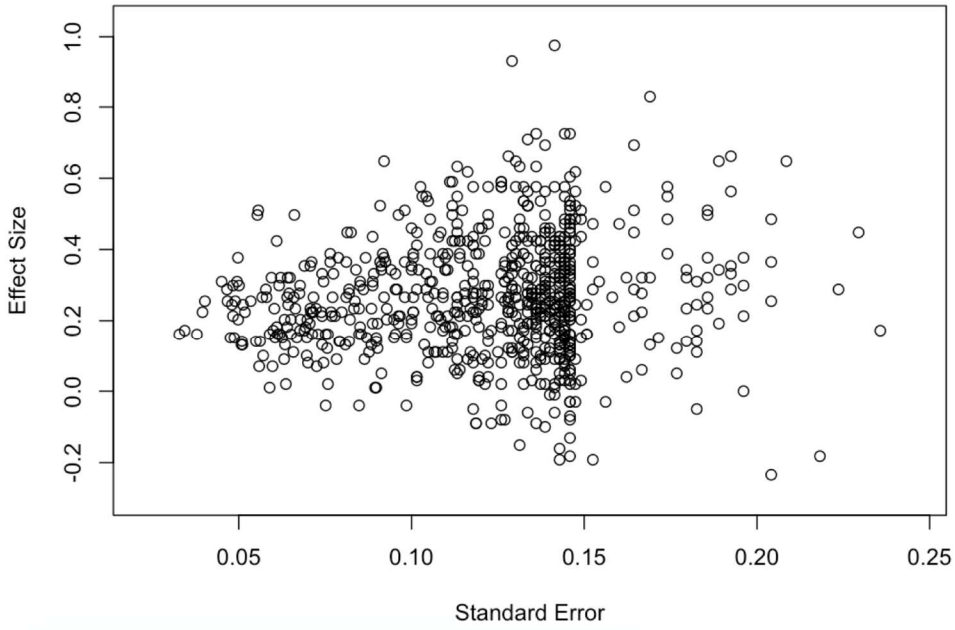


Figure 48. A funnel plot of the GATB dataset in weightr.

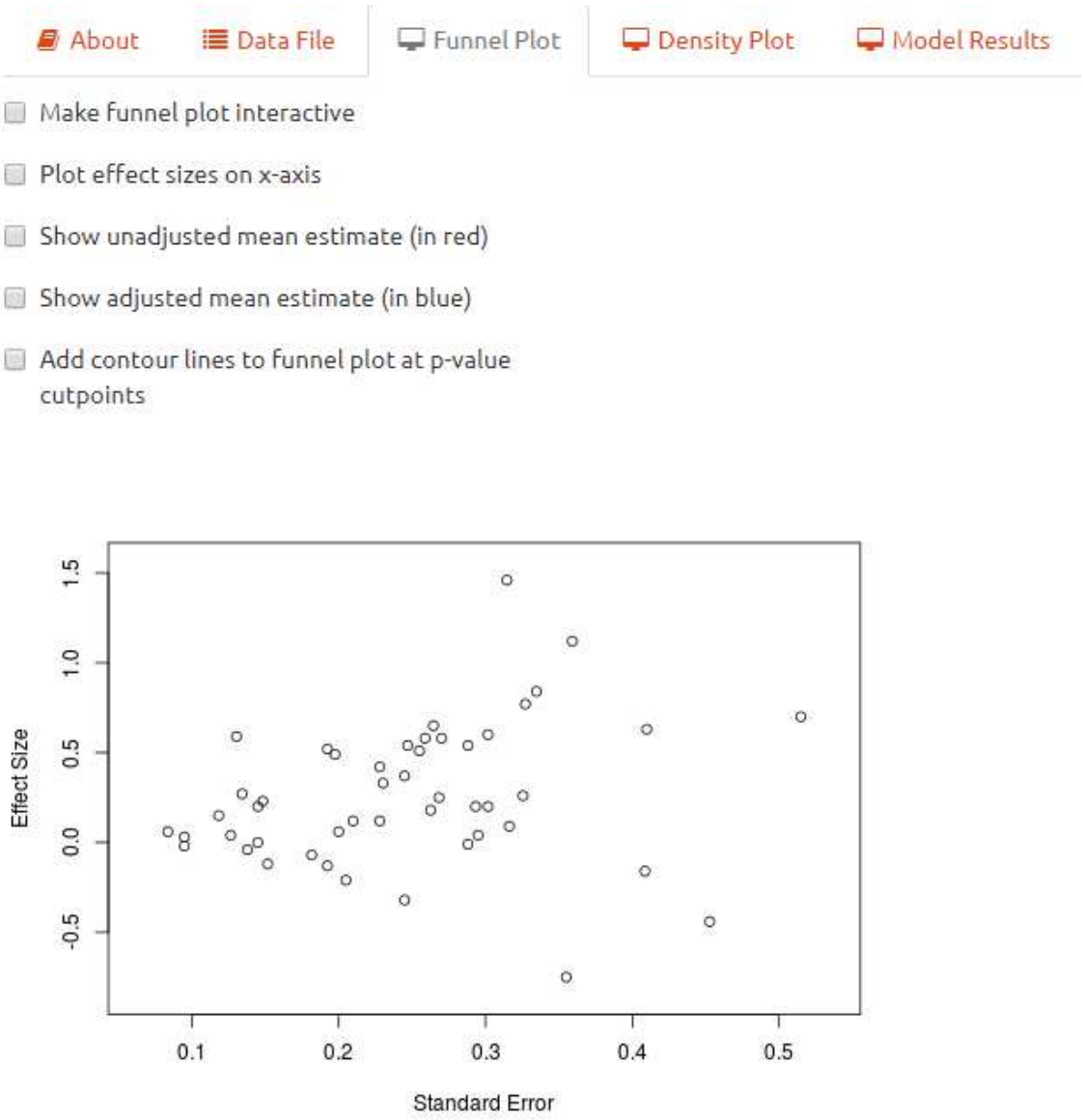
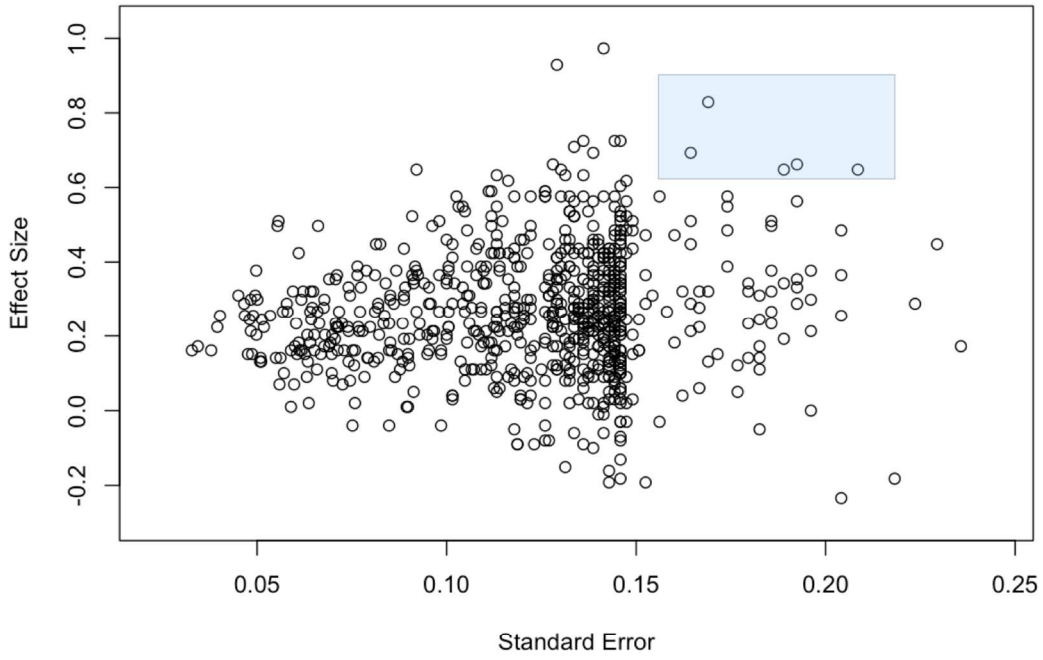


Figure 49. A funnel plot of the Bangert-Drowns dataset in weightr.

Note that the funnel plot *weightr* produces differs from conventional funnel plots. Rather than plotting effect sizes on the horizontal axis, *weightr* plots them on the vertical axis, resulting in the appearance of a horizontal funnel rather than a vertical one. This change is done partly for theoretical reasons and partly for practical ones. Theoretically, when one constructs a scatterplot, the usual convention is to associate the random variable with the vertical axis and the fixed variable with the horizontal; in the case of a funnel plot, effect sizes are the random variable of interest. Therefore, in keeping with graphical traditions, a vertical funnel plot is fundamentally incorrect. In addition, in terms of practicality, a vertical funnel plot can be hard to read, particularly in situations where many small studies are clustered in one area. However, I recognize that vertical funnel plots are widely used, and users who prefer them can check the box labeled "Plot effect sizes on *x*-axis."

There are other checkboxes present above the funnel plot as well; users can opt to plot a line at the unadjusted or adjusted mean effect sizes, or to add contour lines at whatever *p*-value cutpoints they have specified, which results in a contour-enhanced funnel plot. Interested users may also turn the plot into an interactive tool by checking "Make funnel plot interactive," allowing them to identify the *x*- and *y*-values or *p*-values corresponding to a specific point or set of points. Figure 50, featuring the GATB data, is an example of this tool:



If you add contour lines to this plot, they will be drawn at your specified p-value cutpoints -- that is, a cutpoint at 0.05 will draw a 95% confidence interval, one at 0.10 will draw a 90%, and so on. If you have specified a lot of cutpoints, this may be confusing; you can always modify the cutpoints, but keep in mind that your model results will be affected as well.

The following panel gives you information about the funnel plot. If you click on a point, double-click on it, hover over it, or highlight a range of points (brush), that information will appear below.

```
click: x=0.2 y=0.9
dblclick: x=0.1 y=1
hover: pval=0.44
brush: xmin=0.2 xmax=0.2 ymin=0.6 ymax=0.9
```

Figure 50. An interactive funnel plot of the GATB dataset in weightr.

By clicking, double-clicking, clicking and dragging ("brushing"), or hovering, users can select up to four different arrangements of points and view the values

corresponding to those points at once. Hovering over a point yields its *p*-value, while the others yield *x*- and *y*-coordinates; the point with a *p*-value of 0.44 is indicated with a red square.

Users will also find two sliders below the non-interactive funnel plot, one for height and one for width (both measured in pixels). These sliders exist because the point-and-click interface will constrict or expand the plot automatically along with the boundaries of the window or web page. Although this may sometimes be helpful, it can be a hindrance when trying to assess a plot. The sliders allow users to specify their ideal dimensions.

Let us return to Figure 48 and Figure 49 and assess the original funnel plots, as was our intent. For the GATB dataset, the density of the plot is not constant throughout; there appear to be fewer smaller effect sizes, which may be a sign of bias. For the Bangert-Drowns dataset, there is also a drop-off in density around the effect size of zero, and a lack of symmetry. However, as this dataset is small, the change in density may be an artifact of sample size.

The other unique feature of interest in the Shiny application is the density plot. If the user clicks on the tab labeled "Density Plot," they will see a graph much like Figure 51, below:

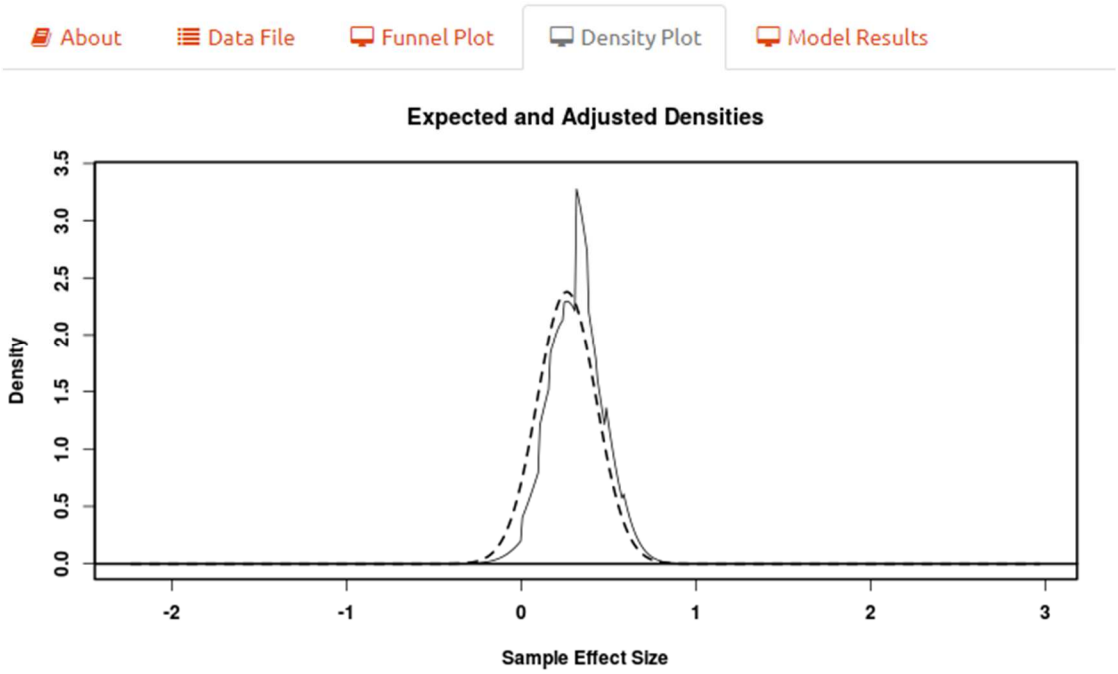


Figure 51. A density plot of the GATB dataset in weightr (the Shiny application).

The density plot provides a graphical representation of the adjustment performed by the weight-function model. If no publication bias is present, the effect sizes are assumed to be normally distributed, with a mean equal to their unadjusted mean and a variance equal to their unadjusted variance component (plus their typical sampling variance). This unadjusted density is depicted by the dashed line. The solid line, on the other hand, depicts the *adjusted* density, where the expected density for effect sizes within each given *p*-value interval is multiplied by the estimated weight for the

corresponding interval. Greater density in an area, therefore, represents a greater likelihood of effect-size survival. (Remember, of course, that the weight for the first interval is fixed to one, and other intervals should be interpreted relative to it.)

Each "bump" in the solid line, then, represents a *p*-value cutpoint. Beginning from the far right, which corresponds to the first *p*-value interval, users can see a slight decrease (the shift from a weight of 1.00 to a weight of 0.84), another decrease (to a weight of 0.69), and so on. Users may wonder why the solid line, representing the adjusted density, falls outside of the unadjusted density, despite the reduced weights. In answer, recall that the mean and variance of the adjusted density differ as well. In this case, the variance-component estimate was adjusted upward, so it is perfectly logical that the adjusted density might be wider and, therefore, might sometimes fall outside its unadjusted counterpart.

These are the features of the Shiny application that pertain to data visualization. For model estimation, we now turn to the other aspect of *weightr*, the *R* function *weightfunct()*. Note that the Shiny application can implement these models as well; users need only select and/or input relevant choices in the sidebar, much as they select effect sizes and sampling variances.

7.1.3 Model Estimation (Vevea and Hedges, 1995)

For the details of interval selection, refer to discussions of the process in Chapters 1, 3, and 4. Assessing the funnel plot can aid in selecting intervals; it may be useful, for instance, to include *p*-values that correspond with areas of changes in density. Including $p = 0.50$ may be of interest; it is the point at which many effect-size metrics become negative.

The GATB data we are using were originally analyzed in Vevea, Clements, and Hedges (1993); we can use the same cutpoints to replicate their analysis. These cutpoints are $p = 0.001$, 0.005, 0.01, 0.02, 0.05, 0.10, 0.20, 0.30, 0.50, and 1.00; they were chosen in part due to their psychological relevance and in part to obtain approximately equal numbers of observed effect sizes per interval (Vevea, Clements, & Hedges, 1993). The size of the GATB dataset is one of its major benefits; it contains so many effect sizes that we can specify a large number of *p*-value cutpoints.

To estimate the model on the GATB data in the *R* console, users can implement the *R* function weightfunct(). For GATB, this code might look something like: weightfunct(effect = dat.gatb$z, v = dat.gatb$v). Effect sizes and variances are the only two arguments that are absolutely required, because, by default, the program uses one cutpoint at $p = 0.025$ (which corresponds to a two-tailed alpha level of 0.05). With the GATB data, to specify more cutpoints, we enter: weightfunct(effect = dat.gatb$z, v = dat.gatb$v, steps = c(0.001, 0.005, 0.01, 0.02, 0.05, 0.10, 0.20, 0.30, 0.50)). (Note that it is not necessary to enter the cutpoints in numerical order, as the function will sort them; nor is it necessary to manually include a cutpoint at $p = 1.00$, or one at $p = 0.00$.)

Running this command estimates both aspects of the Vevea and Hedges (1995) model – an unadjusted model that corresponds to the traditional fixed-effect, mixed-effects, or random-effects meta-analytic model, and a bias-adjusted model. The output will resemble that featured in Figure 52.

```
> weightfunct(effect = dat.gatb$z, v = dat.gatb$v, steps = c(0.001, 0.005, 0.01,
0.02, 0.05, 0.10, 0.20, 0.30, 0.50))

Unadjusted Model (k = 755):

tau^2 (estimated amount of total heterogeneity): 0.0109 (SE = 0.0012)
tau (square root of estimated tau^2 value):  0.1043

Model Results:

          estimate std.error z-stat      p-val ci.lb  ci.ub
Intercept   0.2602  0.005717  45.51 < 2.22e-16 0.249 0.2714

Adjusted Model (k = 755):

tau^2 (estimated amount of total heterogeneity): 0.0176 (SE = 0.0032)
tau (square root of estimated tau^2 value):  0.1327

Model Results:

                 estimate std.error z-stat      p-val     ci.lb  ci.ub
Intercept          0.1903   0.02749  6.923 4.4211e-12  0.136431 0.2442
0.001 < p < 0.005  0.8330   0.12848  6.484 8.9500e-11  0.581192 1.0848
0.005 < p < 0.01   0.6883   0.14039  4.903 9.4548e-07  0.413137 0.9635
0.01 < p < 0.02    0.7216   0.14639  4.929 8.2441e-07  0.434720 1.0086
0.02 < p < 0.05    0.8524   0.17071  4.993 5.9418e-07  0.517791 1.1870
0.05 < p < 0.1     0.5670   0.13591  4.172 3.0222e-05  0.300614 0.8334
0.1 < p < 0.2      0.5290   0.14250  3.712 0.00020523  0.249732 0.8083
0.2 < p < 0.3      0.4493   0.14462  3.107 0.00189031  0.165872 0.7328
0.3 < p < 0.5      0.3136   0.11463  2.736 0.00621787  0.088961 0.5383
0.5 < p < 1        0.1683   0.08738  1.925 0.05417515 -0.003017 0.3395

Likelihood Ratio Test:
X^2(df = 9) = 24.11583, p-val = 0.0041219
```

Figure 52. The Vevea and Hedges (1995) model estimated on the GATB data.

The function provides a lot of information, which can be visually overwhelming. The estimates of the mean (or intercept) and variance component will likely be of primary interest, both for the unadjusted and adjusted models, as will their corresponding standard errors. The likelihood-ratio test may also be informative. The estimates for the weights provide information about the pattern of publication bias, although individually they may be more difficult to interpret.

The estimate of the mean has been reduced from 0.2602 to 0.1903 – an attenuation of about 27%. The variance component has increased from 0.0109 to 0.0176. The likelihood-ratio test is significant, with $df = 9$ and $p < 0.05$. This indicates that the adjusted model is a better fit for the data. Finally, note that the weights for the *p*-value intervals generally decrease moving from *p* of 0.001 to *p* of 1, indicating that effect sizes with larger *p*-values are less likely to survive selection. For this data, it appears that publication bias may be present.

The command table=TRUE will cause the output to include a table of observed effect sizes and *p*-value intervals. weightfunct(effect = dat.gatb$z, v = dat.gatb$v, steps = c(0.001, 0.005, 0.01, 0.02, 0.05, 0.10, 0.20, 0.30, 0.50), table=TRUE) yields Figure 53.

```
Likelihood Ratio Test:
X^2(df = 9) = 24.11583, p-val = 0.0041219

Number of Effect Sizes per Interval:

                          Frequency
p-values <0.001                 214
0.001 < p-values < 0.005         89
0.005 < p-values < 0.01          43
0.01 < p-values < 0.02           55
0.02 < p-values < 0.05          107
0.05 < p-values < 0.1            66
0.1 < p-values < 0.2             72
0.2 < p-values < 0.3             40
0.3 < p-values < 0.5             38
0.5 < p-values < 1               31
```

Figure 53. A table of observed effect sizes and *p*-value intervals based on the GATB data.

The table command is useful in practice; when meta-analysts are selecting a series of *p*-value cutpoints for the Vevea and Hedges (1995) model, they will find the model has difficulty converging if there are few (or no) observed effect sizes in an interval. (Logically, of course, it is impossible to estimate a parameter with no data.) For the GATB example, with the selected cutpoints described above, all intervals contain observed effect sizes, so ideally there will be no problems with estimation.

Another feature of interest is the command fe=TRUE, which forces the software to estimate fixed-effect models. This is especially useful in the case of a "border condition," or circumstances in which the variance component is very near zero and difficult to estimate. The package default is fe=FALSE, a random-effects (or mixed-effects) model. Entering the line: weightfunct(effect = dat.gatb$z, v = dat.gatb$v, steps = c(0.001, 0.005, 0.01, 0.02, 0.05, 0.10, 0.20, 0.30, 0.50), fe=TRUE) yields Figure 54.

```
> weightfunct(effect = dat.gatb$z, v = dat.gatb$v, steps = c(0.001, 0.005,
 0.01, 0.02, 0.05, 0.10, 0.20, 0.30, 0.50), fe=TRUE)

Unadjusted Model (k = 755):

Model Results:

          estimate std.error z-stat      p-val  ci.lb  ci.ub
Intercept   0.2491  0.003729   66.8 < 2.22e-16 0.2418 0.2564

Adjusted Model (k = 755):

Model Results:

                  estimate std.error z-stat      p-val  ci.lb  ci.ub
Intercept           0.2350  0.006302 37.294 < 2.22e-16 0.2227 0.2474
0.001 < p < 0.005   0.6010  0.087287  6.885 5.7692e-12 0.4299 0.7721
0.005 < p < 0.01    0.4749  0.090202  5.265 1.4049e-07 0.2981 0.6517
0.01 < p < 0.02     0.5041  0.092061  5.476 4.3486e-08 0.3237 0.6846
0.02 < p < 0.05     0.6254  0.104406  5.990 2.0998e-09 0.4208 0.8300
0.05 < p < 0.1      0.4615  0.089869  5.136 2.8096e-07 0.2854 0.6377
0.1 < p < 0.2       0.5088  0.103605  4.911 9.0536e-07 0.3058 0.7119
0.2 < p < 0.3       0.5394  0.129522  4.165 3.1198e-05 0.2855 0.7933
0.3 < p < 0.5       0.5118  0.130225  3.930 8.4856e-05 0.2566 0.7671
0.5 < p < 1         0.6782  0.194255  3.491 0.00048066 0.2975 1.0589

Likelihood Ratio Test:
X^2(df = 9) = 30.00153, p-val = 0.00043846
```

Figure 54. A fixed-effect version of the Vevea and Hedges (1995) model, estimated on the GATB data.

Note that information about the variance component is now missing for both models. Estimating a fixed-effect model ignores the between-studies heterogeneity that is present in the data and yields different estimates of the mean and weights. However, if the variance component were zero or near zero, there would be no difference between the fixed-effect and random-effects model results.

To analyze the Bangert-Drowns data, we will need to select a smaller number of cutpoints, because there are fewer effect sizes. However, the Bangert-Drowns dataset has an interesting feature of its own; it includes several potential moderators of effect size. First, let's estimate a random-effects meta-analytic model, without moderators, to choose a set of *p*-value cutpoints.

Running the command weightfunct(effect = dat.bangertdrowns2004$yi, v = dat.bangertdrowns2004$vi, table=TRUE) yields Figure 55.

```
> weightfunct(effect = dat.bangertdrowns2004$yi, v = dat.bangertdrowns2004$vi,
 table=TRUE)

Unadjusted Model (k = 48):

tau^2 (estimated amount of total heterogeneity): 0.0471 (SE = 0.0220)
tau (square root of estimated tau^2 value):  0.2169

Model Results:

          estimate std.error z-stat      p-val ci.lb  ci.ub
Intercept   0.2207   0.04628  4.769 1.8518e-06  0.13 0.3114

Adjusted Model (k = 48):

tau^2 (estimated amount of total heterogeneity): 0.0276 (SE = 0.0238)
tau (square root of estimated tau^2 value):  0.1660

Model Results:

              estimate std.error z-stat    p-val     ci.lb  ci.ub
Intercept       0.1477   0.07311  2.020 0.043428  0.004358 0.2909
0.025 < p < 1   0.4664   0.34924  1.336 0.181690 -0.218065 1.1509

Likelihood Ratio Test:
X^2(df = 1) = 1.163544, p-val = 0.28073

Number of Effect Sizes per Interval:

                     Frequency
p-values <0.025             14
0.025 < p-values < 1        34
```

Figure 55. The Vevea and Hedges (1995) model estimated on the Bangert-Drowns data, two cutpoints.

There are 34 observed *p*-values greater than $p = 0.025$, so we can likely add several cutpoints. The command weightfunct(effect = dat.bangertdrowns2004$yi, v = dat.bangertdrowns2004$vi, table=TRUE, steps=c(0.025, 0.05, 0.10, 0.50)) yields Figure 56.

```
> weightfunct(effect = dat.bangertdrowns2004$yi, v = dat.bangertdrowns2004$vi, table
=TRUE, steps=c(0.025, 0.05, 0.10, 0.50))

Unadjusted Model (k = 48):

tau^2 (estimated amount of total heterogeneity): 0.0471 (SE = 0.0220)
tau (square root of estimated tau^2 value):  0.2169

Model Results:

          estimate std.error z-stat      p-val ci.lb  ci.ub
Intercept   0.2207   0.04628  4.769 1.8518e-06  0.13 0.3114

Adjusted Model (k = 48):

tau^2 (estimated amount of total heterogeneity): 0.0082 (SE = 0.0277)
tau (square root of estimated tau^2 value):  0.0904

Model Results:

                estimate std.error z-stat   p-val   ci.lb  ci.ub
Intercept        0.02913   0.08394 0.3470 0.72861 -0.1354 0.1936
0.025 < p < 0.05 0.19760   0.22351 0.8841 0.37665 -0.2405 0.6357
0.05 < p < 0.1   0.33602   0.41015 0.8193 0.41264 -0.4679 1.1399
0.1 < p < 0.5    0.13140   0.21327 0.6161 0.53782 -0.2866 0.5494
0.5 < p < 1      0.08526   0.15353 0.5553 0.57867 -0.2157 0.3862

Likelihood Ratio Test:
X^2(df = 4) = 3.799974, p-val = 0.43375

Number of Effect Sizes per Interval:

                       Frequency
p-values <0.025               14
0.025 < p-values < 0.05        2
0.05 < p-values < 0.1          6
0.1 < p-values < 0.5          15
0.5 < p-values < 1            11
Warning message:
In weightfunct(effect = dat.bangertdrowns2004$yi, v = dat.bangertdrowns2004$vi,  :
  At least one of the p-value intervals contains three or fewer effect sizes, which
may lead to estimation problems. Consider re-specifying the cutpoints.
```

Figure 56. Attempting to estimate too many *p*-value cutpoints with the Vevea and Hedges (1995) model.

The package produces a warning when there are three or fewer observed effects in a given *p*-value interval. (Three is an arbitrary small number; the warning does not *guarantee* that an estimation problem has occurred, merely alerts the user to the possibility.) There are only 2 observed effects in one interval and 6 in another. If we remove the cutpoint at $p = 0.05$, things may look more reasonable. The command weightfunct(effect = dat.bangertdrowns2004$yi, v = dat.bangertdrowns2004$vi, table=TRUE, steps=c(0.025, 0.10, 0.50)) yields Figure 57.

```
> weightfunct(effect = dat.bangertdrowns2004$yi, v = dat.bangertdrowns2004$vi, table
=TRUE, steps=c(0.025, 0.10, 0.50))

Unadjusted Model (k = 48):

tau^2 (estimated amount of total heterogeneity): 0.0471 (SE = 0.0220)
tau (square root of estimated tau^2 value):  0.2169

Model Results:

          estimate std.error z-stat      p-val ci.lb  ci.ub
Intercept   0.2207   0.04628  4.769 1.8518e-06  0.13 0.3114

Adjusted Model (k = 48):

tau^2 (estimated amount of total heterogeneity): 0.0036 (SE = 0.0180)
tau (square root of estimated tau^2 value):  0.0601

Model Results:

                estimate std.error z-stat   p-val    ci.lb  ci.ub
Intercept        0.01886   0.05462 0.3452 0.72994 -0.08820 0.1259
0.025 < p < 0.1  0.23724   0.20130 1.1785 0.23858 -0.15730 0.6318
0.1 < p < 0.5    0.09832   0.11644 0.8444 0.39844 -0.12989 0.3265
0.5 < p < 1      0.06270   0.07864 0.7973 0.42528 -0.09144 0.2168

Likelihood Ratio Test:
X^2(df = 3) = 3.390182, p-val = 0.33529

Number of Effect Sizes per Interval:

                       Frequency
p-values <0.025               14
0.025 < p-values < 0.1         8
0.1 < p-values < 0.5          15
0.5 < p-values < 1            11
```

Figure 57. The Vevea and Hedges (1995) model estimated on the Bangert-Drowns data with adequate cutpoints.

Now no intervals contain three or fewer observed effects, and we can see that the mean estimate (under intercept) has been adjusted downward from 0.2207 to 0.0189 – a drastic change. Effect sizes with $p > 0.50$ are, in fact, only about 0.06 times as likely to survive as effect sizes with $p < 0.025$. Although the likelihood-ratio test is nonsignificant ($p = 0.34$), possibly due to the smaller number of effects and the limited number of cutpoints, the drastic reduction in effect size is informative. In practice, users should try out a few sets of cutpoints and observe the resulting fluctuation in estimates. If the effect estimate is resilient and does not change much, the data are likely to be robust to possible publication bias. In this case, however, the estimate was reduced so far that publication bias is likely to be a threat to this dataset.

Now we can explore some of the moderators in this dataset. For this example, we will work with the variables *length* and *grade*. Recall that the Bangert-Drowns data studies the effectiveness of school-based writing-to-learn interventions on academic achievement. *Length* is a continuous moderator representing the length of the intervention in number of weeks. *Grade* is a categorical moderator representing the grade during which the intervention was administered, with four levels (1 = elementary, 2 = middle, 3 = high school, 4 = college). The values of *grade* are numeric, but the variable is categorical, so we need to redefine the variable and tell *R* that it is a factor. To do so, we

use the command grade <- as.factor(dat.bangertdrowns2004$grade). We also redefine *length* by mean-centering it: length <- (dat.bangertdrowns2004$length - mean(dat.bangertdrowns2004$length, na.rm=TRUE)). We usually mean-center continuous variables in contexts where a score of zero on the variable is difficult to comprehend; mean-centering is also advantageous when calculating interaction terms.

To run the mixed-effects model, we enter the command: weightfunct(effect = dat.bangertdrowns2004$yi, v = dat.bangertdrowns2004$vi, steps = c(0.025, 0.10, 0.50), mods = ~length + grade). This yields Figure 58.

```
> weightfunct(effect=dat.bangertdrowns2004$yi, v=dat.bangertdrowns2004$vi, steps=c(
0.025, 0.10, 0.50), mods =~length+grade)

Unadjusted Model (k = 46):

tau^2 (estimated amount of total heterogeneity): 0.0372 (SE = 0.0173)
tau (square root of estimated tau^2 value):  0.1928

Model Results:

           estimate std.error   z-stat     p-val     ci.lb   ci.ub
Intercept  0.222223  0.082643  2.68896 0.0071676  0.060246 0.38420
length     0.012869  0.007486  1.71904 0.0856073 -0.001804 0.02754
grade2    -0.261698  0.165555 -1.58073 0.1139399 -0.586181 0.06278
grade3     0.106933  0.132551  0.80673 0.4198208 -0.152862 0.36673
grade4    -0.006268  0.106716 -0.05873 0.9531664 -0.215426 0.20289

Adjusted Model (k = 46):

tau^2 (estimated amount of total heterogeneity): 0.0232 (SE = 0.0209)
tau (square root of estimated tau^2 value):  0.1524

Model Results:

                estimate std.error  z-stat   p-val     ci.lb   ci.ub
Intercept        0.10043  0.109847  0.9143 0.36056 -0.114864 0.31573
length           0.01073  0.007444  1.4417 0.14938 -0.003858 0.02532
grade2          -0.21869  0.149912 -1.4588 0.14462 -0.512510 0.07513
grade3           0.07673  0.124777  0.6150 0.53858 -0.167826 0.32129
grade4          -0.02820  0.096813 -0.2912 0.77087 -0.217945 0.16155
0.025 < p < 0.1  0.56503  0.417357  1.3538 0.17579 -0.252971 1.38304
0.1 < p < 0.5    0.32300  0.318568  1.0139 0.31062 -0.301377 0.94739
0.5 < p < 1      0.20785  0.254257  0.8175 0.41366 -0.290489 0.70618

Likelihood Ratio Test:
X^2(df = 3) = 2.182011, p-val = 0.5355
There were  2 cases removed from your dataset due to the presence of missing data.
To view the row numbers of these cases, use the attribute '$removed'.
```

Figure 58. The Vevea and Hedges (1995) model estimated on the Bangert-Drowns data with moderators of effect size.

The unadjusted intercept, after including these two moderator variables, is now $d = 0.22$; it represents the average effect of school-based writing-to-learn programs on academic achievement when the length of the program is 9.83 weeks (the average length) and the program is administered in elementary school (level 1 of *grade*).

The unadjusted *length* parameter estimate is approximately $d = 0.013$, which indicates that for every week longer the program is, the effect on academic achievement increases by approximately 0.013 standard deviations.

For levels 2, 3, and 4 of the *grade* variable (representing middle school, high school, and college, respectively), we can calculate the unadjusted conditional means by summing the intercept and each of the respective parameter estimates. The conditional mean for middle school programs is 0.2222223 – 0.261698 = -0.039475, indicating that such programs in middle school *reduce* academic achievement, albeit only by about 0.04 standard deviations. The conditional mean for college is less than that for middle school programs as well, although there is still an improvement in academic achievement (d = 0.216). The largest improvement in academic achievement occurs in high school, where the unadjusted conditional mean is d = 0.329.

The adjusted intercept (the effect of elementary school programs that are 9.83 weeks long) is reduced by about 50%, to d = 0.100. The *length* parameter estimate hasn't changed too much; it has moved from d = 0.013 to d = 0.011. The conditional mean for middle school programs has been adjusted downward, from d = -0.039 to d = -0.1183. For high school programs, it has been adjusted from d = 0.329 to d = 0.17716. Finally, for college programs, it has moved from d = 0.216 to d = 0.07223.

In this case, the estimates for grade levels are reduced more than the estimate for length. This certainly can happen and is, in fact, one of the advantages of this model; the model does not restrict conditions to be adjusted in the same direction or to the same degree. The effect of grade level appears to be more vulnerable to publication bias than the effect of length.

The model provides a notification at the bottom, informing users that two cases were removed from the dataset due to the presence of missing data. The command weightfunct(effect = dat.bangertdrowns2004$yi, v = dat.bangertdrowns2004$vi, steps = c(0.025, 0.10, 0.50), mods = ~length + grade)$removed tells us that those cases were row numbers 34 and 35. We can view those cases with the command dat.bangertdrowns2004[34:35,], which results in Figure 59.

```
> dat.bangertdrowns2004[34:35,]
   id    author year grade length minutes wic feedback info pers imag meta    subject ni   yi    vi
34 34 Nieswandt 1997     3     NA      NA   0        1    1    0    0    0  Chemistry 91 0.12 0.044
35 35 Radmacher 1995     4     NA      NA   0        1    1    0    0    1 Psychology 36 1.12 0.129
```

Figure 59. Cases with missing data removed by weightr.

These two cases were removed because they had no scores, or scores of *NA*, on the *length* variable.

Researchers who use the model (or models) for simulation purposes will likely want to extract the model results and store them. To do so, save the output of weightfunct( ) as an object (here called *wf_out*). Then commands such as wf_out$unadj_par will produce a vector of the unadjusted estimates, and so forth. The command wf_out[1] extracts the first element of a list, which prints out the more technical output of the unadjusted model (see Figure 60).

```
> wf_out[1]
[[1]]
[[1]]$par
[1]  0.037152947  0.222223186  0.012869477 -0.261698132  0.106932965 -0.006267511

[[1]]$value
[1] -30.76468

[[1]]$counts
function gradient
      48       48

[[1]]$convergence
[1] 0

[[1]]$message
[1] "CONVERGENCE: REL_REDUCTION_OF_F <= FACTR*EPSMCH"

[[1]]$hessian
           [,1]       [,2]        [,3]       [,4]       [,5]       [,6]
[1,] 3653.69814 -266.00764  -941.52080   59.84677  -65.87835 -143.78886
[2,] -266.00764  534.33914  -190.85657   56.46595  102.07441  218.42945
[3,] -941.52080 -190.85657 20692.75804 -245.57629 -259.69558  -78.82871
[4,]   59.84677   56.46595  -245.57629   56.46595    0.00000    0.00000
[5,]  -65.87835  102.07441  -259.69558    0.00000  102.07441    0.00000
[6,] -143.78886  218.42945   -78.82871    0.00000    0.00000  218.42945
```

Figure 60. Extracting the Hessian matrix from weightr.

From this, users can extract the Hessian matrix, which contains all the information necessary to calculate the standard error for each parameter, as well as corresponding *z*-tests and *p*-values. (This process is not described here for the sake of brevity.)

I have described the process of estimating the Vevea and Hedges (1995) model. I now move on to the Vevea and Woods (2005) model, followed by the lambda model variations.

7.1.4 Model Estimation (Vevea and Woods, 2005)

To estimate the Vevea and Woods (2005) model, users need only add the single argument *weights* to weightfunct( ). This argument allows users to input a pre-specified vector of weights for their corresponding *p*-value cutpoints.

For the GATB example, we ran weightfunct(effect = dat.gatb$z, v = dat.gatb$v, steps = c(0.001, 0.005, 0.01, 0.02, 0.05, 0.10, 0.20, 0.30, 0.50)) to estimate the model. We can specify weights for those *p*-value intervals. There are 10 intervals, so we must specify 10 weights. The command weightfunct(effect = dat.gatb$z, v = dat.gatb$v, steps = c(0.001, 0.005, 0.01, 0.02, 0.05, 0.10, 0.20, 0.30, 0.50), weights=c(1, 1, 1, 1, 1, 0.80, 0.70, 0.60, 0.50, 0.10)) results in Figure 61.

```
> weightfunct(effect = dat.gatb$z, v = dat.gatb$v, steps = c(0.001, 0.005, 0.01,
 0.02, 0.05, 0.10, 0.20, 0.30, 0.50), weights=c(1, 1, 1, 1, 1, 0.80, 0.70, 0.60,
 0.50, 0.10))

Unadjusted Model (k = 755):

tau^2 (estimated amount of total heterogeneity): 0.0109 (SE = 0.0012)
tau (square root of estimated tau^2 value):  0.1043

Model Results:

          estimate std.error z-stat      p-val ci.lb  ci.ub
Intercept   0.2602  0.005717  45.51 < 2.22e-16 0.249 0.2714

Adjusted Model (k = 755):

tau^2 (estimated amount of total heterogeneity): 0.0233 (SE = ---)
tau (square root of estimated tau^2 value):  0.1526

Model Results:

                         estimate std.error z-stat p-val ci.lb ci.ub
Intercept         0.187291543088105       ---    ---   ---   ---   ---
0.001 < p < 0.005                 1       ---    ---   ---   ---   ---
0.005 < p < 0.01                  1       ---    ---   ---   ---   ---
0.01 < p < 0.02                   1       ---    ---   ---   ---   ---
0.02 < p < 0.05                   1       ---    ---   ---   ---   ---
0.05 < p < 0.1                  0.8       ---    ---   ---   ---   ---
0.1 < p < 0.2                   0.7       ---    ---   ---   ---   ---
0.2 < p < 0.3                   0.6       ---    ---   ---   ---   ---
0.3 < p < 0.5                   0.5       ---    ---   ---   ---   ---
0.5 < p < 1                     0.1       ---    ---   ---   ---   ---

Note: The symbol --- appears because the user has specified weights,
choosing to use the Vevea and Woods model, which does not estimate
weights for p-value intervals and therefore cannot produce meaningful
standard errors. The likelihood ratio test is also not interpretable.
```

Figure 61. The Vevea and Woods (2005) model estimated on the GATB data.

This pattern of weights represents a case in which all effect sizes with *p*-values below 0.05 survive selection, and the chance of survival drops from there until those with *p*-values above 0.50 are only 10% as likely to survive, relative to significant effects. This is a strong pattern of publication bias, which reduces the overall effect size from 0.26 to 0.19.

A lot of information is absent in Figure 61 – no standard errors are provided, and everything based on the standard error is absent as well, including confidence intervals. The note explains why; the weights are not actually being estimated, so their standard errors are not meaningful. It is theoretically possible to calculate a standard error for the intercept, because that parameter is estimated, but the estimate is based purely on user-specified information, so its standard error will not be meaningful either. The Vevea and Woods (2005) model is designed to allow users to test out a variety of selection patterns and observe their effect, rather than to estimate a pattern from the data.

The rest of the function operates as before. Users can calculate a fixed-effect model; they can include moderators, request a table of *p*-value cutpoints, extract results, and so on. The two important things to remember about the *weights* argument are that the user must specify the same number of weights as there are *p*-value intervals and that the

weights must be specified in the same order as the intervals they correspond to. (The latter is a good reason for specifying the *p*-value cutpoints in numerical order.)

We now move to estimation of the lambda model, the focus of this dissertation.

7.1.5 Model Estimation (Lambda model; Coburn and Vevea, in prep)

The lambda model is not yet implemented in the Shiny application, but that update is forthcoming, along with its public release.

To estimate this model and its variants, we will switch to the Bem et al. (2016) dataset, because neither the GATB nor the Bangert-Drowns dataset contains a moderator across which publication bias is likely to vary. To create a dummy-coded variable distinguishing the earlier studies from the later ones, we enter the command dummy <- c(rep(1, length(early$ES)), rep(0, length(later$ES))), creating a vector of ones with length equal to that of the early effect sizes and zeroes with length equal to that of the later effects.

To estimate the standard lambda model, exactly as described in Chapter 4 (a fixed-effect model with the same cutpoints), we run the command: weightfunct(effect = effects, v = variances, steps = c(0.025, 0.05, 0.10, 0.50, 0.90, 1.00), lambda_model = dummy, fe = TRUE). We set the argument lambda_model equal to the dummy-coded variable representing the relevant study characteristic. The output is featured in Figure 62.

```
> weightfunct(effect = effects, v = variances, steps = c(.025, .05, .10,
  .50, .90, 1), lambda_model = dummy, fe = TRUE)

Unadjusted Model (k = 78):

Model Results:

          estimate std.error z-stat      p-val   ci.lb   ci.ub
Intercept   0.0527  0.009498  5.548 2.8892e-08 0.03408 0.07131

Adjusted Model (k = 78):

Model Results:

                estimate std.error z-stat    p-val     ci.lb   ci.ub
Intercept        0.02283   0.01513  1.509 0.131307 -0.006823 0.05248
0.025 < p < 0.05 0.86589   0.39382  2.199 0.027901  0.094013 1.63777
0.05 < p < 0.1   0.47020   0.26764  1.757 0.078947 -0.054367 0.99477
0.1 < p < 0.5    0.24681   0.13440  1.836 0.066299 -0.016607 0.51024
0.5 < p < 0.9    0.28793   0.18818  1.530 0.125986 -0.080886 0.65675
0.9 < p < 1      0.26683   0.23848  1.119 0.263193 -0.200582 0.73424
Lambda           0.54729   0.29048  1.884 0.059554 -0.022042 1.11663

Likelihood Ratio Test:
X^2(df = 6) = 18.43063, p-val = 0.0052415
```

Figure 62. The lambda model (in prep) estimated on the Bem data.

Note that the results correspond exactly to those described in Chapter 4.

For the Vevea and Woods (2005) variant of the lambda model, as described in Chapter 5, we simply add the *weights* command. We match the pattern of selection described as "extreme two-tailed selection" in Chapter 5. The code is as follows: weightfunct(effect = effects, v = variances, steps = c(0.005, 0.01, 0.05, 0.10, 0.25, 0.35, 0.50, 0.65, 0.75, 0.90, 0.95, 0.99, 0.95, 1), weights = c(1, 1, 1, .50, .25, .10, .10, .10, .10, .25, .50, 1, 1, 1), fe = TRUE, lambda_model = dummy). It yields Figure 63.

```
> weightfunct(effect = effects, v = variances, steps = c(.005, .010, .050, .10
0, .250, .350, .500, .650, .750, .900, .950, .990, .995, 1), weights = c(1, 1,
 1, .50, .25, .10, .10, .10, .10, .25, .50, 1, 1, 1), fe = TRUE, lambda_model
= dummy)

Unadjusted Model (k = 78):

Model Results:

          estimate std.error z-stat      p-val   ci.lb   ci.ub
Intercept   0.0527  0.009498  5.548 2.8892e-08 0.03408 0.07131

Adjusted Model (k = 78):

Model Results:

                         estimate std.error z-stat p-val ci.lb ci.ub
Intercept        0.0254840366494902       ---    ---   ---   ---   ---
0.005 < p < 0.01                  1       ---    ---   ---   ---   ---
0.01 < p < 0.05                   1       ---    ---   ---   ---   ---
0.05 < p < 0.1                  0.5       ---    ---   ---   ---   ---
0.1 < p < 0.25                 0.25       ---    ---   ---   ---   ---
0.25 < p < 0.35                 0.1       ---    ---   ---   ---   ---
0.35 < p < 0.5                  0.1       ---    ---   ---   ---   ---
0.5 < p < 0.65                  0.1       ---    ---   ---   ---   ---
0.65 < p < 0.75                 0.1       ---    ---   ---   ---   ---
0.75 < p < 0.9                 0.25       ---    ---   ---   ---   ---
0.9 < p < 0.95                  0.5       ---    ---   ---   ---   ---
0.95 < p < 0.99                   1       ---    ---   ---   ---   ---
0.99 < p < 0.995                  1       ---    ---   ---   ---   ---
0.995 < p < 1                     1       ---    ---   ---   ---   ---
Lambda            0.761525056155744       ---    ---   ---   ---   ---

Note: The symbol --- appears because the user has specified weights,
choosing to use the Vevea and Woods model, which does not estimate
weights for p-value intervals and therefore cannot produce meaningful
standard errors. The likelihood ratio test is also not interpretable.
```

Figure 63. The lambda model as sensitivity analysis (in prep) estimated on the Bem data.

Note that the results correspond exactly to those described in Chapter 5.

Although detailed examples are not provided here for the sake of space, it is of course possible to combine all the arguments described above with the *lambda_model* argument, including moderators of effect size, tables of *p*-values, and so on. It is also possible to extract parameter estimates in the same fashion as described.

7.2 Conclusions

I hope that this chapter aids interested users of the Vevea and Hedges (1995) model and its variations, including the lambda model. Using the *weightr* package, it is possible to replicate the analyses and simulations described in this dissertation. The package is already available for free through the Comprehensive R Archive Network (or CRAN), and it will include the lambda model once the model is released.

A model is useless if it cannot be implemented. In the time since the release of *weightr*, the package has been downloaded thousands of times and cited several times. I believe that it will continue to see use, and that the Vevea and Hedges (1995) class of models – including the lambda model – will gain popularity and traction as a result.

Chapter 8: Discussion

This dissertation begins in Chapter 1 by outlining the pervasive problem of publication bias. In Chapter 2, the issue is refined to that of study characteristics, such as funding source, and their influence on publication bias; evidence from various fields is provided. Chapter 3 describes the simulation used to evaluate the models included in the dissertation, and Chapters 4, 5, and 6 present several implementations of a new model, referred to as the lambda model, to address the role of study characteristics. Chapter 4 implements a standard maximum-likelihood model, Chapter 5 implements a variant with fixed weights, and Chapter 6 demonstrates a Bayesian version. Finally, Chapter 7 acknowledges the necessity of software for model implementation, and presents a tutorial on my *R* package, *weightr*, to implement various weight-function models.

There is no quick, simple cure for publication bias, no easy method of eliminating future bias or adjusting for its impact on all existing scientific research. However, we can always strive to improve. To eliminate sources of bias in future research, we can encourage the pre-registration of studies; we can ensure that our analyses are replicable and our data are available, as I have done in the Appendices of this dissertation. To adjust past studies' results, we must understand that *p*-values are not the only factor impacting the likelihood of publication.

The lambda model is not perfect – by necessity, it makes some restrictive assumptions – but it is an effective start. The simulation results reveal that the lambda model, in all its variants, tends to do a better job of reproducing the population mean than an unadjusted meta-analytic model when any degree of bias is present. In future research, I aim to conduct additional simulations on the performance of the lambda model. In addition, and perhaps more importantly, I will continue to dedicate my time and energy to this worthwhile cause – the quest to understand and eliminate publication bias in scientific literature. I sincerely hope that other researchers will join me, creating their own models and posing their own theories. In this way, we will improve the "science" of science.

References

Abdel‐Sattar, M., Krauth, D., Anglemyer, A., & Bero, L. (2014). The relationship between risk of bias criteria, research outcomes, and study sponsorship in a cohort of preclinical thiazolidinedione animal studies: a meta‐analysis. *Evidence-based preclinical medicine*, *1*(1), 11-20.

Bangert-Drowns, R. L., Hurley, M. M., & Wilkinson, B. (2004). The effects of school-based writing-to-learn interventions on academic achievement: A meta-analysis. *Review of educational research*, *74*(1), 29-58.

Banks, G. C., Kepes, S., & Banks, K. P. (2012). Publication bias: The antagonist of meta-analytic reviews and effective policymaking. *Educational Evaluation and Policy Analysis*, *34*(3), 259-277.

Barnes, D. E., & Bero, L. A. (1998). Why review articles on the health effects of passive smoking reach different conclusions. *Jama*, *279*(19), 1566-1570.

Becker, B. J. (2005). Failsafe N or file-drawer number. *Publication bias in meta-analysis: Prevention, assessment and adjustments*, 111-125.

Begg, C. B., & Berlin, J. A. (1988). Publication bias: a problem in interpreting medical data. *Journal of the Royal Statistical Society. Series A (Statistics in Society)*, 419-463.

Begg, C. B., & Berlin, J. A. (1989). Publication bias and dissemination of clinical research. *JNCI: Journal of the National Cancer Institute*, *81*(2), 107-115.

Begg, C. B., & Mazumdar, M. (1994). Operating characteristics of a rank correlation test for publication bias. *Biometrics*, 1088-1101.

Bem, D. J. (1972). Self-perception theory. In *Advances in experimental social psychology* (Vol. 6, pp. 1-62). Academic Press.

Bem, D. J. (2000). Exotic becomes erotic: Interpreting the biological correlates of sexual orientation. *Archives of Sexual Behavior*, *29*(6), 531-548.

Bem, D. J. (2011). Feeling the future: experimental evidence for anomalous retroactive influences on cognition and affect. *Journal of personality and social psychology*, *100*(3), 407.

Bem, D., Tressoldi, P., Rabeyron, T., & Duggan, M. (2016). Feeling the future: A meta-analysis of 90 experiments on the anomalous anticipation of random future events.

Berlin, J. A., Begg, C. B., & Louis, T. A. (1989). An assessment of publication bias using a sample of published clinical trials. *Journal of the American Statistical Association*, *84*(406), 381-392.

Borenstein, M., Hedges, L. V., Higgins, J. P., & Rothstein, H. R. (2011). *Introduction to meta-analysis*. John Wiley & Sons.

Carpenter, J. R., Schwarzer, G., Rücker, G., & Künstler, R. (2009). Empirical evaluation showed that the Copas selection model provided a useful summary in 80% of meta-analyses. *Journal of clinical epidemiology*, *62*(6), 624-631.

Carter, C. (Producer, creator). (1993-2002). *The X-Files*. [Television series.] Vancouver, Canada: Fox.

Chalmers, I., Hedges, L. V., & Cooper, H. (2002). A brief history of research synthesis. *Evaluation & the health professions*, *25*(1), 12-37.

Clarke, M., Brice, A., & Chalmers, I. (2014). Accumulating research: a systematic account of how cumulative meta-analyses would have provided knowledge, improved health, reduced harm and saved resources. *PLoS One*, *9*(7), e102670.

Coburn, K. M. & Vevea, J. L. (2015). Publication bias as a function of study characteristics. *Psychological Methods*, *20*(3), 310-330.

Coburn, K. M. & Vevea, J. L. (2017). weightr: Estimating weight-function models for publication bias. [R package.]

Coburn, K. M. & Vevea, J. L. (in prep). Modeling moderators of publication bias.

Coburn, K. M., Vevea, J. L., & Orey, B. (in prep). A survey of the number of studies in meta-analyses across publications.

Cohen, J. (1988). Statistical power analysis for the behavioral sciences. 2nd edition.

Cooper, H., Hedges, L. V., & Valentine, J. C. (Eds.). (2009). *The handbook of research synthesis and meta-analysis*. Russell Sage Foundation.

Copas, J. B., & Li, H. G. (1997). Inference for Non‐random Samples. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *59*(1), 55-95.

Copas, J., & Shi, J. Q. (2000). Meta-analysis, funnel plots and sensitivity analysis. *Biostatistics*, *1*(3), 247-262.

Copas, J. B., & Shi, J. Q. (2001). A sensitivity analysis for publication bias in systematic reviews. *Statistical methods in medical research*, *10*(4), 251-265.

Cumming, G. (2013). *Understanding the new statistics: Effect sizes, confidence intervals, and meta-analysis*. Routledge.

Davidson, R. A. (1986). Source of funding and outcome of clinical trials. *Journal of General Internal Medicine*, *1*(3), 155-158.

Dickersin, K., & Min, Y. I. (1993a). NIH clinical trials and publication bias.

Dickersin, K., & Min, Y. I. (1993b). Publication bias: the problem that won't go away. *Annals of the New York Academy of Sciences*, *703*(1), 135-148.

Djulbegovic, B., Lacevic, M., Cantor, A., Fields, K. K., Bennett, C. L., Adams, J. R., ... & Lyman, G. H. (2000). The uncertainty principle and industry-sponsored research. *The Lancet*, *356*(9230), 635-638.

Duval, S., & Tweedie, R. (2000a). Trim and fill: a simple funnel‐plot–based method of testing and adjusting for publication bias in meta‐analysis. *Biometrics*, *56*(2), 455-463.

Duval, S., & Tweedie, R. (2000b). A nonparametric "trim and fill" method of accounting for publication bias in meta-analysis. *Journal of the American Statistical Association*, *95*(449), 89-98.

Easterbrook, P. J., Gopalan, R., Berlin, J. A., & Matthews, D. R. (1991). Publication bias in clinical research. *The Lancet*, *337*(8746), 867-872.

Egger, M., Smith, G. D., Schneider, M., & Minder, C. (1997). Bias in meta-analysis detected by a simple, graphical test. *Bmj*, *315*(7109), 629-634.

Etter, J. F., Burri, M., & Stapleton, J. (2007). The impact of pharmaceutical company funding on results of randomized trials of nicotine replacement therapy for smoking cessation: a meta‐analysis. *Addiction*, *102*(5), 815-822.

Fanelli, D. (2012). Negative results are disappearing from most disciplines and countries. *Scientometrics*, *90*(3), 891-904.

Ferguson, C. J., & Brannick, M. T. (2012). Publication bias in psychological science: prevalence, methods for identifying and controlling, and implications for the use of meta-analyses. *Psychological methods*, *17*(1), 120.

Fernández-i-Marín, X. (2016). ggmcmc: Analysis of MCMC samples and Bayesian inference. *Journal of Statistical Software*, *70*(9).

Gelfand, A. E., & Smith, A. F. (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American statistical association*, *85*(410), 398-409.

Gelman, A. (1995). Inference and Monitoring Convergence in Markov Chain Monte Carlo in Practice. WR Gilks, S. Richardson, and DJ Spiegelhalter, eds.

Gelman, A. (1996). Bayesian model-building by pure thought: Some principles and examples. *Statistica Sinica*, 215-232.

Gelman, A. (2006). Prior distributions for variance parameters in hierarchical models (comment on article by Browne and Draper). *Bayesian analysis*, *1*(3), 515-534.

Gelman, A. (2009). Prior distributions for Bayesian data analysis in political science.

Gilks, W. R., Richardson, S., & Spiegelhalter, D. (Eds.). (1995). *Markov chain Monte Carlo in practice*. CRC press.

Gelman, A., & Rubin, D. B. (1992a). Inference from iterative simulation using multiple sequences. *Statistical science*, 457-472.

Gelman, A., & Rubin, D. B. (1992b). A single series from the Gibbs sampler provides a false sense of security. *Bayesian statistics*, *4*, 625-631.

Geweke, J. (1991). *Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments*(Vol. 196). Minneapolis, MN, USA: Federal Reserve Bank of Minneapolis, Research Department.

Geyer, C. J. (1992). Practical markov chain monte carlo. *Statistical science*, 473-483.

Grégoire, G., Derderian, F., & Le Lorier, J. (1995). Selecting the language of the publications included in a meta-analysis: is there a Tower of Babel bias?. *Journal of clinical epidemiology*, *48*(1), 159-163.

Hedges, L. V. (1984). Estimation of effect size under nonrandom sampling: The effects of censoring studies yielding statistically insignificant mean differences. *Journal of Educational Statistics*, *9*(1), 61-85.

Hedges, L. V., & Vevea, J. L. (1998). Fixed-and random-effects models in meta-analysis. *Psychological methods*, *3*(4), 486.

Heidelberger, P., & Welch, P. D. (1981). A spectral method for confidence interval generation and run length control in simulations. *Communications of the ACM*, *24*(4), 233-245.

Heidelberger, P., & Welch, P. D. (1983). Simulation run length control in the presence of an initial transient. *Operations Research*, *31*(6), 1109-1144.

Higgins, J., & Thompson, S. G. (2002). Quantifying heterogeneity in a meta‐analysis. *Statistics in medicine*, *21*(11), 1539-1558.

Hopewell, S., Clarke, M. J., Stewart, L., & Tierney, J. (2007). Time to publication for results of clinical trials. *The Cochrane Library*.

Hunter, J. P., Saratzis, A., Sutton, A. J., Boucher, R. H., Sayers, R. D., & Bown, M. J. (2014). In meta-analyses of proportion studies, funnel plots were found to be an

inaccurate method of assessing publication bias. *Journal of clinical epidemiology*, *67*(8), 897-903.
Huss, A., Egger, M., Hug, K., Huwiler-Müntener, K., & Röösli, M. (2008). Source of funding and results of studies of health effects of mobile phone use: systematic review of experimental studies. *Ciencia & saude coletiva*, *13*(3), 1005-1012.
Hyde, J. S., & Linn, M. C. (1988). Gender differences in verbal ability: A meta-analysis. *Psychological bulletin*, *104*(1), 53.
Ialongo, C. (2016). Understanding the effect size and its measures. *Biochemia medica: Biochemia medica*, *26*(2), 150-163.
Ioannidis, J. P. (1998). Effect of the statistical significance of results on the time to completion and publication of randomized efficacy trials. *Jama*, *279*(4), 281-286.
Iyengar, S., & Greenhouse, J. B. (1988). Selection models and the file drawer problem. *Statistical Science*, 109-117.
Izrailev, S. (2014). Tictoc: Functions for timing R scripts, as well as implementations of stack and list structures. [R package.]
Jasra, A., Holmes, C. C., & Stephens, D. A. (2005). Markov chain Monte Carlo methods and the label switching problem in Bayesian mixture modeling. *Statistical Science*, 50-67.
Kepes, S., Banks, G. C., McDaniel, M., & Whetzel, D. L. (2012). Publication bias in the organizational sciences. *Organizational Research Methods*, *15*(4), 624-662.
Kepes, S., Banks, G. C., & Oh, I. S. (2014). Avoiding bias in publication bias research: The value of "null" findings. *Journal of Business and Psychology*, *29*(2), 183-203.
King, W. R., & He, J. (2006). A meta-analysis of the technology acceptance model. *Information & management*, *43*(6), 740-755.
Kjaergard, L., & Als-Nielsen, B. (2002). Association between competing interests and authors' conclusions: epidemiological study of randomised clinical trials published in the BMJ. *Bmj*, *325*(7358), 249.
Kruschke, J. K., Aguinis, H., & Joo, H. (2012). The time has come: Bayesian methods for data analysis in the organizational sciences. *Organizational Research Methods*, *15*(4), 722-752.
Lane, D. M., & Dunlap, W. P. (1978). Estimating effect size: Bias resulting from the significance criterion in editorial decisions. *British Journal of Mathematical and Statistical Psychology*, *31*(2), 107-112.
Lau, J., Ioannidis, J. P., Terrin, N., Schmid, C. H., & Olkin, I. (2006). Evidence based medicine: The case of the misleading funnel plot. *BMJ: British Medical Journal*, *333*(7568), 597.
LeLorier, J., Gregoire, G., Benhaddad, A., Lapierre, J., & Derderian, F. (1997). Discrepancies between meta-analyses and subsequent large randomized, controlled trials. *New England Journal of Medicine*, *337*(8), 536-542.
Lesser, L. I., Ebbeling, C. B., Goozner, M., Wypij, D., & Ludwig, D. S. (2007). Relationship between funding source and conclusion among nutrition-related scientific articles. *PLoS Medicine*, *4*(1), e5.
Lexchin, J., Bero, L. A., Djulbegovic, B., & Clark, O. (2003). Pharmaceutical industry sponsorship and research outcome and quality: systematic review. *Bmj*, *326*(7400), 1167-1170.

Light, R. J. Pillemer, DB. (1984). *Summing up: The science of reviewing research.* Cambridge.

Lipsey, M. W., & Wilson, D. B. (2001). *Practical meta-analysis*. Sage Publications, Inc.

Lunn, D., Jackson, C., Best, N., Thomas, A., & Spiegelhalter, D. (2012). *The BUGS book: A practical introduction to Bayesian analysis*. CRC press.

Macaskill, P., Walter, S. D., & Irwig, L. (2001). A comparison of methods to detect publication bias in meta‐analysis. *Statistics in medicine*, *20*(4), 641-654.

McCambridge, J., & Hartwell, G. (2015). Has industry funding biased studies of the protective effects of alcohol on cardiovascular disease? A preliminary investigation of prospective cohort studies. *Drug and alcohol review*, *34*(1), 58-66.

Melander, H., Ahlqvist-Rastad, J., Meijer, G., & Beermann, B. (2003). Evidence b (i) ased medicine—selective reporting from studies sponsored by pharmaceutical industry: review of studies in new drug applications. *Bmj*, *326*(7400), 1171-1173.

Microsoft Corporation and Weston, S. (2017). doParallel: foreach parallel adaptor for the 'parallel' package. [R package.]

Nichols, H. (1891). The psychology of time. *The American Journal of Psychology*, *3*(4), 453-529.

Orwin, R. G. (1983). A fail-safe N for effect size in meta-analysis. *Journal of educational statistics*, *8*(2), 157-159.

Parekh-Bhurke, S., Kwok, C. S., Pang, C., Hooper, L., Loke, Y. K., Ryder, J. J., ... & Song, F. (2011). Uptake of methods to deal with publication bias in systematic reviews has increased over time, but there is still much scope for improvement. *Journal of clinical epidemiology*, *64*(4), 349-357.

Peters, J. L., Sutton, A. J., Jones, D. R., Abrams, K. R., & Rushton, L. (2006). Comparison of two methods to detect publication bias in meta-analysis. *Jama*, *295*(6), 676-680.

Plummer, M. (2003). JAGS: A program for analysis of Bayesian graphical models using Gibbs sampling.

R Core Team (2017). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Raftery, A. E., & Banfield, J. D. (1991). Stopping the Gibbs Sampler, the Use of Morphology, and Other Issues in Spatial Statistics (Bayesian image restoration, with two applications in spatial statistics) -- (Discussion). *Annals of the Institute of Statistical Mathematics*, *43*(1), p32-43.

Raftery, A. E., & Lewis, S. M. (1992). [Practical Markov Chain Monte Carlo]: comment: one long run with diagnostics: implementation strategies for Markov Chain Monte Carlo. *Statistical science*, *7*(4), 493-497.

Rice, K., Higgins, J., & Lumley, T. (2018). A re‐evaluation of fixed effect (s) meta‐analysis. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, *181*(1), 205-227.

Rosenthal, R. (1979). The file drawer problem and tolerance for null results. *Psychological bulletin*, *86*(3), 638.

Rosenthal, R., & DiMatteo, M. R. (2001). Meta-analysis: Recent developments in quantitative methods for literature reviews. *Annual review of psychology*, *52*(1), 59-82.

Rothstein, H. R. (2008). Publication bias as a threat to the validity of meta-analytic results. *Journal of Experimental Criminology*, *4*(1), 61-81.

Rothstein, H. R., Sutton, A. J., & Borenstein, M. (Eds.). (2006). *Publication bias in meta-analysis: Prevention, assessment and adjustments*. John Wiley & Sons.

Scargle, J. D. (1999). Publication bias (the" file-drawer problem") in scientific inference. *arXiv preprint physics/9909033*.

Schmidt, F. L., & Hunter, J. E. (2014). *Methods of meta-analysis: Correcting error and bias in research findings*. Sage publications.

Scholz, F. W. (1985). Maximum likelihood estimation. In Encyclopedia of Statistical Sciences 5, (S. Kotz, N. L. Johnson, and C. B. Read, eds.). New York: Wiley.

Scifres, C. M., Iams, J. D., Klebanoff, M., & Macones, G. A. (2009). Metaanalysis vs large clinical trials: which should guide our management?. *American journal of obstetrics and gynecology*, *200*(5), 484-e1.

Simpson, R. J. S., & Pearson, K. (1904). Report on certain enteric fever inoculation statistics. *The British Medical Journal*, 1243-1246.

Sismondo, S. (2008). Pharmaceutical company funding and its consequences: a qualitative systematic review. *Contemporary Clinical Trials*, *29*(2), 109-113.

Slate (2017, May 17). Daryl Bem proved ESP is real (which means science is broken). [Blog post]. Retrieved from https://slate.com/health-and-science/2017/06/daryl-bem-proved-esp-is-real-showed-science-is-broken.html

Smith, M. L., Glass, G. V., & Miller, T. I. (1980). *The benefits of psychotherapy*. Baltimore: Johns Hopkins University Press.

Sterne, J. A., & Egger, M. (2001). Funnel plots for detecting bias in meta-analysis: guidelines on choice of axis. *Journal of clinical epidemiology*, *54*(10), 1046-1055.

Sterne, J. A., Egger, M., & Smith, G. D. (2001). Systematic reviews in health care: Investigating and dealing with publication and other biases in meta-analysis. *BMJ: British Medical Journal*, *323*(7304), 101.

Stigler, S. M. (1989). Francis Galton's account of the invention of correlation. *Statistical Science*, 73-79.

Tang, J. L., & Liu, J. L. (2000). Misleading funnel plot for detection of bias in meta-analysis. *Journal of clinical epidemiology*, *53*(5), 477-484.

Terrin, N., Schmid, C. H., & Lau, J. (2005). In an empirical evaluation of the funnel plot, researchers could not visually identify publication bias. *Journal of clinical epidemiology*, *58*(9), 894-901.

Thornton, A., & Lee, P. (2000). Publication bias in meta-analysis: its causes and consequences. *Journal of clinical epidemiology*, *53*(2), 207-216.

Tsung-han T., Gill, J., & Rapkin, J. (2012). superdiag: R code for testing Markov chain nonconvergence. [R package.]

van de Schoot, R., & Depaoli, S. (2014). Bayesian analyses: Where to start and what to report. *European Health Psychologist*, *16*(2), 75-84.

Vevea, J. L., Clements, N. C., & Hedges, L. V. (1993). Assessing the effects of selection bias on validity data for the General Aptitude Test Battery. *Journal of Applied Psychology*, *78*(6), 981.

Vevea, J. L. & Coburn, K. M. (in press). Publication bias. In Cooper, H., Hedges, L. V., & Valentine, J. C. (Eds.), *Handbook of Research Synthesis and Meta-Analysis*. Russell Sage Foundation.

Vevea, J. L., & Hedges, L. V. (1995). A general linear model for estimating effect size in the presence of publication bias. *Psychometrika*, *60*(3), 419-435.

Vevea, J. L., & Woods, C. M. (2005). Publication bias in research synthesis: sensitivity analysis using a priori weight functions. *Psychological methods*, *10*(4), 428.

Vevea, J. L., Zelinsky, N., Turitz Mitchell, M., Castaneda, R., & Coburn, K. M. (in prep). Distributions of sample sizes in meta-analyses across distributions.

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. Journal of Statistical Software, 36(3), 1-48. URL: http://www.jstatsoft.org/v36/i03/

Wickham, H. (2009). ggplot2: Elegant graphics for data analysis. Springer-Verlag, New York.

Yu-Sung, S. and Masanao, Y. (2015). R2jags: Using R to run 'JAGS.' [R package.]

Zhang, Z., Hamagami, F., Lijuan Wang, L., Nesselroade, J. R., & Grimm, K. J. (2007). Bayesian analysis of longitudinal data using growth curve models. *International Journal of Behavioral Development*, *31*(4), 374-383.

Appendix A: Bem Dataset

The *early* variable here is coded 0 for studies published after 2011, 1 for studies published before 2011, and 2 for studies published in 2011. A PDF of the meta-analysis is available at the following URL: https://www.researchgate.net/profile/Patrizio_Tressoldi/publication/304843991_Feeling_the_future_A_meta-analysis_of_90_experiments_on_the_anomalous_anticipation_of_random_future_events/links/5884ea3eaca272b7b44a858e/Feeling-the-future-A-meta-analysis-of-90-experiments-on-the-anomalous-anticipation-of-random-future-events.pdf

	Year	Task	N	ES	SE	Early
1	2013	word recall	102	-0.054	0.098	0
2	2009	priming	120	0.093	0.091	1
3	2008	habituation	43	0.139	0.151	1
4	2010	habituation	70	0.205	0.119	1
5	2008	priming	50	0.471	0.147	1
6	2009	habituation	46	0.268	0.148	1
7	2011	reward	100	0.249	0.101	2
8	2011	avoidance	150	0.194	0.082	2
9	2011	priming	97	0.257	0.102	2
10	2011	priming	99	0.202	0.101	2
11	2011	habituation	100	0.221	0.1	2
12	2011	habituation	150	0.145	0.082	2
13	2011	habituation	200	0.092	0.071	2
14	2011	word recall	100	0.191	0.1	2
15	2011	word recall	50	0.412	0.145	2
16	2012	reward	42	0.285	0.155	0
17	2011	priming	169	0.108	0.077	2
18	2013	retro-practice	67	0.255	0.123	0
19	2006	priming	51	0.411	0.144	1
20	2009	word recall	38	-0.043	0.159	1
21	2012	reward	59	0.145	0.129	0
22	2012	retro-practice	194	0.139	0.072	0
23	2012	retro-practice	416	0.061	0.049	0
24	2012	word	112	-0.113	0.094	0

		recall word				
25	2012	recall word	158	0	0.079	0
26	2012	recall word	124	0.114	0.09	0
27	2012	recall word	109	0.168	0.096	0
28	2012	recall word	211	-0.049	0.069	0
29	2012	recall word	106	-0.029	0.096	0
30	2012	recall word	2469	-0.005	0.02	0
31	2005	habituation	47	0.085	0.144	1
32	2012	reward	50	-0.05	0.139	0
33	2012	reward	52	0.044	0.137	0
34	2012	reward	50	0.159	0.14	0
35	2012	reward	49	0.228	0.142	0
36	2009	reward	41	0.182	0.155	1
37	2008	reward	100	0.249	0.101	1
38	2008	reward	25	0.504	0.206	1
39	2008	reward	32	0.347	0.178	1
40	2011	word recall	88	0.026	0.106	2
41	2012	avoidance	63	-0.012	0.124	0
42	2012	avoidance	406	-0.024	0.05	0
43	2012	avoidance	111	0.251	0.096	0
44	2012	avoidance	201	0.21	0.071	0
45	2012	avoidance	1222	0.068	0.029	0
46	2012	avoidance	327	0.1	0.055	0
47	2013	avoidance	640	0.052	0.04	0
48	2010	word recall	58	-0.012	0.13	1
49	2004	habituation	40	0.313	0.159	1
50	2001	habituation	72	0.178	0.118	1
51	2004	habituation	183	-0.01	0.074	1
52	2004	habituation	203	-0.06	0.07	1
53	2004	habituation	203	0.07	0.07	1
54	2003	priming	68	-0.129	0.121	1
55	2010	habituation	20	0.249	0.218	1
56	2012	word recall	96	0.186	0.102	0
57	2012	word recall	98	0.111	0.101	0
58	2012	habituation	50	-0.142	0.14	0
59	2014	priming	28	-0.248	0.187	0

60	2009	priming word	155	0.106	0.08	1
61	2012	recall word	50	0.015	0.139	0
62	2012	recall word	50	-0.219	0.141	0
63	2012	recall word	50	-0.04	0.139	0
64	2011	recall	50	-0.118	0.14	2
65	2012	priming word	47	0.099	0.144	0
66	2012	recall	50	0.078	0.139	0
67	2012	priming word	42	-0.096	0.152	0
68	2012	recall	50	-0.042	0.139	0
69	2003	habituation	84	0.17	0.109	1
70	2002	priming	40	0.128	0.156	1
71	2002	priming	50	0.166	0.14	1
72	2002	priming	54	0	0.134	1
73	2004	habituation	25	0.329	0.199	1
74	2005	habituation	50	0.284	0.142	1
75	2005	habituation word	92	-0.018	0.103	1
76	2013	recall	52	0.049	0.137	0
77	2009	habituation word	50	-0.163	0.14	1
78	2013	recall word	75	0.279	0.117	0
79	2013	recall word	25	0.292	0.198	0
80	2013	recall	26	-0.399	0.198	0
81	2012	text speed	48	0.06	0.142	0
82	2012	text speed	60	-0.249	0.129	0
83	2012	priming word	100	0.036	0.099	0
84	2012	recall	100	0.221	0.1	0
85	2012	reward word	103	0.12	0.098	0
86	2012	recall	104	-0.007	0.097	0
87	2013	retro-practice	102	0.152	0.099	0
88	2012	reward	100	-0.022	0.099	0
89	2000	reward	60	-0.076	0.128	1
90	2006	habituation	52	0.046	0.137	1

Appendix B: R Simulation Code

The following code, with appropriate variations as necessary and where indicated, is sufficient to replicate all simulation cells presented in this dissertation.

```
## Create a directory for the given cell:
setwd("K:/")
dir.create("Cell 72")
setwd("K:/Cell 72/")
start_time <- Sys.time()

## Packages only need to be installed once, so the next lines can be commented out later.
install.packages("weightr")
install.packages("R2jags")
install.packages("tictoc")
install.packages("doParallel")

library(weightr)
library(R2jags)
library(tictoc)
library(doParallel)

source("C:/Users/kcobu/Desktop/Sample Size Functions.R")

## These lines set up parallel processing. Number of cores may vary depending on computing power. (I used 15.)
detectCores()
cl <- makeCluster(15)
registerDoParallel(cl)
```

```
## This is the likelihood function for the adjusted lambda model.
neglike3 <- function(pars) {
  vc = pars[1]
  mn = XX %*% pars[2]
  w = c(1, pars[3])
  lambda = pars[4]

  eta = sqrt(v + vc)

  contrib = log(w[wt])
  for (i in 1:length(dummy)) {
    if (wt[i] > 1) {
      if (dummy[i] == 1 && steps[wt[i]] > .05) {
        contrib[i] = log(lambda * w[wt[i]])
      }
    }
  }
  a = sum(contrib)
  b = 1 / 2 * sum(((effect - mn) / eta) ^ 2)
  c = sum(log(eta))
  Bij <- matrix(rep(0, number * nsteps), nrow = number, ncol = nsteps)
  bi = -si * qnorm(steps[1])
  Bij[, 1] = 1 - pnorm((bi - mn) / eta)
  if (nsteps > 2) {
    for (j in 2:(length(steps) - 1)) {
      bi = -si * qnorm(steps[j])
      bilast = -si * qnorm(steps[j - 1])
      Bij[, j] = pnorm((bilast - mn) / eta) - pnorm((bi - mn) / eta)
    }
  }
  bilast = -si * qnorm(steps[length(steps) - 1])
```

```
  Bij[, length(steps)] = pnorm((bilast - mn) / eta)

  swbij = 0
  for (j in 1:length(steps)) {
    contrib = w[j] * Bij[, j]
    for (i in 1:length(dummy)) {
      if (j > 1) {
        if (dummy[i] == 1 && steps[j] > .05)
          contrib[i] = lambda * w[j] * Bij[i, j]
      }
    }
    swbij = swbij + contrib
  }

  d = sum(log(swbij))
  return(-a + b + c + d)
}

## This is the likelihood function for the Vevea and Woods (2005) version of the lambda model.
neglike4 <- function(pars) {
  vc = pars[1]
  mn = XX %*% pars[2]
  w = weights
  lambda = pars[3]

  eta = sqrt(v + vc)

  contrib = log(w[wt])
  for (i in 1:length(dummy)) {
    if (wt[i] > 1) {
      if (dummy[i] == 1 && steps[wt[i]] > .05) {
```

```
      contrib[i] = log(lambda * w[wt[i]])
    }
  }
}
a = sum(contrib)
b = 1 / 2 * sum(((effect - mn) / eta) ^ 2)
c = sum(log(eta))
Bij <- matrix(rep(0, number * nsteps), nrow = number, ncol = nsteps)
bi = -si * qnorm(steps[1])
Bij[, 1] = 1 - pnorm((bi - mn) / eta)
if (nsteps > 2) {
  for (j in 2:(length(steps) - 1)) {
    bi = -si * qnorm(steps[j])
    bilast = -si * qnorm(steps[j - 1])
    Bij[, j] = pnorm((bilast - mn) / eta) - pnorm((bi - mn) / eta)
  }
}
bilast = -si * qnorm(steps[length(steps) - 1])
Bij[, length(steps)] = pnorm((bilast - mn) / eta)

swbij = 0
for (j in 1:length(steps)) {
  contrib = w[j] * Bij[, j]
  for (i in 1:length(dummy)) {
    if (j > 1) {
      if (dummy[i] == 1 && steps[j] > .05)
        contrib[i] = lambda * w[j] * Bij[i, j]
    }
  }
  swbij = swbij + contrib
}
```

```
  d = sum(log(swbij))
  return(-a + b + c + d)
}

## This section specifies some information. First, the seed:
set.seed(3435)

## Number of reps; note that this is processed in parallel, so it is number of reps per core.
reps <- 667

## The variance component. I2 of 0% corresponds to vc of 0, 25% = 0.03, 50% = 0.08, and 75% = 0.23
## K in each group is total k divided by 2; for 172, 86
vc1 <- 0.03
mu1 <- 0.2
k1 <- 86

vc2 <- 0.03
mu2 <- 0.2
k2 <- 86

start <- 1

   foreach(i = 1:15, .packages = c('weightr', 'R2jags',         'tictoc')) %dopar% {

         while (start < reps) {

           tryCatch({

             for (i in start:reps) {
```

```
tic()

### Data Generation ###

## Uncomment other data generation mechanisms as necessary. All four selection mechanisms are included here. ##

## 1.0
#
# ## Group 0 ##
#
# # For strong:
# w1 <- c(1.0, 0.2)
# # For weak:
# w1 <- c(1.0, 0.7)
# # For none:
w1 <- c(1.0, 1.0)

output1 <- matrix(0,(20*k1),2)

for(j in 1:(20*k1)) {

  n1a <- samplesize_no_outliers(1)
  n1b <- samplesize_no_outliers(1)

  v1 <- ( (n1a + n1b)/(n1a*n1b) ) + ( mu1^2/(2*(n1a + n1b)) )
  d1 <- rnorm(1,mu1,sqrt(v1 + vc1))
  p1 <- 1-pnorm(d1/sqrt(v1))

  pint1 = 1 #p < .05
  if(p1 > .05) pint1 = 2
```

```
  if(runif(1) < w1[pint1]) output1[j, 1:2] = c(d1,v1)

}

output1 <- output1[rowSums(output1) !=0, , drop=TRUE]
d1_survived <- output1[1:k1,1]
v1_survived <- output1[1:k1,2]

# ## Group 1 ##
#
# # w2 <- c(1.0,1.0)
# # For strong:
# w2 <- c(1.0, 0.2)
# # For weak:
w2 <- c(1.0, 0.7)
#
output2 <- matrix(0,(20*k2),2)
#
for(j in 1:(20*k2)) {

  n2a <- samplesize_no_outliers(1)
  n2b <- samplesize_no_outliers(1)

  v2 <- ( (n2a + n2b)/(n2a*n2b) ) + ( mu2^2/(2*(n2a + n2b)) )
  d2 <- rnorm(1,mu2,sqrt(v2 + vc2))
  p2 <- 1-pnorm(d2/sqrt(v2))

  pint2 = 1 #p < .05
  if(p2 > .05) pint2 = 2

  if(runif(1) < w2[pint2]) output2[j, 1:2] = c(d2,v2)
```

```
}

output2 <- output2[rowSums(output2) !=0, , drop=TRUE]
d2_survived <- output2[1:k2,1]
v2_survived <- output2[1:k2,2]

d <- c(d1_survived, d2_survived)
v <- c(v1_survived, v2_survived)
dummy <- c(rep(0,k1),rep(1,k2))

# ## 2.0
#
# ## Group 0 ##
#

 # n1a <- samplesize_no_outliers(20 * k1)
 # n1b <- samplesize_no_outliers(20 * k1)
 #
 # v1 <- ((n1a + n1b) / (n1a * n1b)) + (mu1 ^ 2 / (2 * (n1a + n1b)))
 # d1 <- rnorm(20 * k1, mu1, sqrt(v1 + vc1))
 # #
 # p1 <- 1-pnorm(d1/sqrt(v1))

 # For selection, uncomment as needed.
 # .5 = weak, 2 = strong

 # prob1 <- exp(-.5*p1)
 # y1 <- rbinom(n=length(d1), size=1, prob=prob1)
 # df1 <- data.frame(y=y1, d=d1, v=v1, p=p1, prob=prob1)
 # data1 <- df1[(df1[,1] == 1),]
```

```
# d1_survived <- data1$d[1:k1]
# v1_survived <- data1$v[1:k1]

#  df1 <- data.frame(d = d1, v = v1)
# #
#  d1_survived <- df1$d[1:k1]
#  v1_survived <- df1$v[1:k1]

#
# ## Group 1 ##
#

# n2a <- samplesize_no_outliers(20 * k2)
# n2b <- samplesize_no_outliers(20 * k2)
# #
# v2 <- ((n2a + n2b) / (n2a * n2b)) + (mu2 ^ 2 / (2 * (n2a + n2b)))
# d2 <- rnorm(20 * k2, mu2, sqrt(v2 + vc2))
# #
# p2 <- 1 - pnorm(d2 / sqrt(v2))

# prob2 <- exp(-2 * p2)
# y2 <- rbinom(n = length(d2),
#              size = 1,
#              prob = prob2)
# df2 <- data.frame(
#   y = y2,
#   d = d2,
#   v = v2,
#   p = p2,
#   prob = prob2
# )
```

```
 # data2 <- df2[(df2[, 1] == 1), ]
 # d2_survived <- data2$d[1:k2]
 # v2_survived <- data2$v[1:k2]

  # df2 <- data.frame(d=d2, v=v2)
 #
  # d2_survived <- df2$d[1:k2]
  # v2_survived <- df2$v[1:k2]

 # d <- c(d1_survived, d2_survived)
 # v <- c(v1_survived, v2_survived)
 # dummy <- c(rep(0, k1), rep(1, k2))

# ## 3.0
#
# ## Group 0 ##
# #
 # output1 <- matrix(0,(2000*k1),2)
#
#  # Steps = .1, .3, .5, .7, .9
# w1 <- c(1,1,1,1,1,1)
#  # Strong weights, possibly?
#  #w1 <- c(0.2, 0.2, 0.2, 1, 1, 1)
#  # Weak weights, possibly?
#  w1 <- c(0.7, 0.7, 0.7, 1, 1, 1)
#
#  for(i in 1:2000*k1){
#
#    n1a <- samplesize_no_outliers(1)
#    n1b <- samplesize_no_outliers(1)
#
```

```
#    v1 <- ( (n1a + n1b)/(n1a*n1b) ) + ( mu1^2/(2*(n1a + n1b)) )
#    d1 <- rnorm(1,mu1,sqrt(v1 + vc1))
#
#    ef_int1 = 1 #effect is < .1
#    if(d1 > .1 & d1 <= .3) ef_int1 = 2
#    if(d1 > .3 & d1 <= .5) ef_int1 = 3
#    if(d1 > .5 & d1 <= .7) ef_int1 = 4
#    if(d1 > .7 & d1 <= .9) ef_int1 = 5
#    if(d1 > .9) ef_int1 = 6
#
#    if(runif(1) < w1[ef_int1]) output1[i, 1:2] = c(d1,v1)
#  }
# #
#  output1 <- output1[rowSums(output1) !=0, , drop=TRUE]
#  d1_survived <- output1[1:k1,1]
#  v1_survived <- output1[1:k1,2]
#
#  #
#  # ## Group 1 ##
#  #
 # output2 <- matrix(0,(2000*k2),2)
#
#  # Steps = .1, .3, .5, .7, .9
#  # w2 <- c(1,1,1,1,1,1,1)
#  # Strong weights, possibly?
#  w2 <- c(0.2, 0.2, 0.2, 1, 1, 1)
#  # Weak weights, possibly?
 # w2 <- c(0.7, 0.7, 0.7, 1, 1, 1)
#  # w2 <- c(1,1,1,1,1,1)
#
#
```

```
#  for(i in 1:2000*k2){
#
#    n2a <- samplesize_no_outliers(1)
#    n2b <- samplesize_no_outliers(1)
#
#    v2 <- ( (n2a + n2b)/(n2a*n2b) ) + ( mu2^2/(2*(n2a + n2b)) )
#    d2 <- rnorm(1,mu2,sqrt(v2 + vc2))
#
#    ef_int2 = 1 #effect is < .1
#    if(d2 > .1 & d2 <= .3) ef_int2 = 2
#    if(d2 > .3 & d2 <= .5) ef_int2 = 3
#    if(d2 > .5 & d2 <= .7) ef_int2 = 4
#    if(d2 > .7 & d2 <= .9) ef_int2 = 5
#    if(d2 > .9) ef_int2 = 6
#
#    if(runif(1) < w2[ef_int2]) output2[i, 1:2] = c(d2,v2)
#  }
#
#  output2 <- output2[rowSums(output2) !=0, , drop=TRUE]
#  d2_survived <- output2[1:k2,1]
#  v2_survived <- output2[1:k2,2]
# # # #
#  d <- c(d1_survived, d2_survived)
#  v <- c(v1_survived, v2_survived)
# dummy <- c(rep(0,k1),rep(1,k2))
#
# ## 4
# # Chance of survival is a logistic function of effect size
# #
# # ## Group 0 ##
# #
```

```
# n1a <- samplesize_no_outliers(20 * k1)
# n1b <- samplesize_no_outliers(20 * k1)
#
# v1 <- ((n1a + n1b) / (n1a * n1b)) + (mu1 ^ 2 / (2 * (n1a + n1b)))
# d1 <- rnorm(20 * k1, mu1, sqrt(v1 + vc1))
#
# #prob2 for strong, prob4 for weak
# # prob2 <- exp(5*(d1 - 0.464))/(1 + exp(5*(d1 - 0.464)))
# # prob4 <- exp(3*(d1 - -0.464))/(1 + exp(3*(d1 - -0.464)))
#
# # If no selection, comment out lines 411-416
#
# # y1 <- rbinom(n=length(d1), size=1, prob=prob4)
# # df1 <- data.frame(y=y1, d=d1, v=v1, prob=prob4)
# # data1 <- df1[(df1[,1] == 1),]
#
# # d1_survived <- data1$d[1:k1]
# # v1_survived <- data1$v[1:k1]
#
# # If selection IS present, comment out lines 420-421
#
# d1_survived <- d1[1:k1]
# v1_survived <- v1[1:k1]
#
# # ## Group 1 ##
# #
# n2a <- samplesize_no_outliers(20 * k2)
# n2b <- samplesize_no_outliers(20 * k2)
#
# v2 <- ((n2a + n2b) / (n2a * n2b)) + (mu2 ^ 2 / (2 * (n2a + n2b)))
# d2 <- rnorm(20 * k2, mu2, sqrt(v2 + vc2))
```

```
#
# # prob2 for strong, prob4 for weak -- change other lines
# # prob2 <- exp(5*(d2 - 0.464))/(1 + exp(5*(d2 - 0.464)))
# prob4 <- exp(3*(d2 - -0.464))/(1 + exp(3*(d2 - -0.464)))
#
#
# # If no selection, comment out lines 437-442
#
# y2 <- rbinom(n=length(d2), size=1, prob=prob2)
# df2 <- data.frame(y=y2, d=d2, v=v2, prob=prob2)
# data2 <- df2[(df2[,1] == 1),]
#
# d2_survived <- data2$d[1:k2]
# v2_survived <- data2$v[1:k2]
#
# # If selection IS present, comment out lines 446-447
#
# # d2_survived <- d2[1:k2]
# # v2_survived <- v2[1:k2]
#
# d <- c(d1_survived, d2_survived)
# v <- c(v1_survived, v2_survived)
# dummy <- c(rep(0,k1),rep(1,k2))
#

###### Model Estimation ######

## Unadjusted Random-Effects Model ##

orig_models <- weightfunct(d, v)
unadj_random <- rbind(orig_models[1][[1]]$par)
```

```
write.table(
  unadj_random,
  "unadj_random.csv",
  append = TRUE,
  row.names = FALSE,
  col.names = FALSE,
  sep = ","
)

## Original Vevea and Hedges Model ##

orig_vandh <- rbind(orig_models[2][[1]]$par)
write.table(
  orig_vandh,
  "orig_vandh.csv",
  append = TRUE,
  row.names = FALSE,
  col.names = FALSE,
  sep = ","
)

## Lambda, weights "correctly" specified

steps <- c(.05, 1.0)
npred <- 0
intercept <- TRUE
XX <- matrix(nrow = length(dummy), ncol = 1)
XX[, 1] <- rep(1, length(dummy))
number <- length(dummy)
nsteps <- 2
effect <- d
```

```
si <- sqrt(v)

pars <- c(mean(v) / 4, mean(d), 0.1, 0.5)

wt <- rep(1, number)
for (i in 1:number) {
  for (j in 2:nsteps) {
    if (-si[i] * qnorm(steps[j]) <= d[i] &&
        d[i] <= -si[i] * qnorm(steps[j - 1]))
      wt[i] = j
  }
  if (d[i] <= -si[i] * qnorm(steps[nsteps - 1]))
    wt[i] = nsteps
}

orig_lambda <-
  optim(
    par = pars,
    fn = neglike3,
    lower = c(0, -Inf, 0.01, 0.01),
    method = "L-BFGS-B",
    hessian = TRUE
  )
orig_lambda_pars <- rbind(orig_lambda$par)
write.table(
  orig_lambda_pars,
  "orig_lambda.csv",
  append = TRUE,
  row.names = FALSE,
  col.names = FALSE,
```

```
  sep = ","
)

## Lambda Vevea and Woods (weights fixed, lambda free) Model 1

steps <-
  c(
    0.005,
    0.010,
    0.050,
    0.100,
    0.250,
    0.350,
    0.500,
    0.650,
    0.750,
    0.900,
    0.950,
    0.990,
    0.995,
    1
  )
nsteps <- length(steps)

pars1 <- c(mean(v) / 4, mean(d), 0.5)
weights <-
  c(1,
    0.99,
    0.95,
    0.90,
    0.80,
```

```
    0.75,
    0.65,
    0.60,
    0.55,
    0.50,
    0.50,
    0.50,
    0.50,
    0.50,
    0.50)

wt <- rep(1, number)
for (i in 1:number) {
  for (j in 2:nsteps) {
    if (-si[i] * qnorm(steps[j]) <= d[i] &&
        d[i] <= -si[i] * qnorm(steps[j - 1]))
      wt[i] = j
  }
  if (d[i] <= -si[i] * qnorm(steps[nsteps - 1]))
    wt[i] = nsteps
}

orig_lambda_vw1 <-
  optim(
    par = pars1,
    fn = neglike4,
    lower = c(0, -Inf, 0.01),
    method = "L-BFGS-B",
    hessian = TRUE
  )
orig_lambda_vw1_pars <- rbind(orig_lambda_vw1$par)
write.table(
```

```
  orig_lambda_vw1_pars,
  "orig_lambda_vw1.csv",
  append = TRUE,
  row.names = FALSE,
  col.names = FALSE,
  sep = ","
)

## Lambda Vevea and Woods (weights fixed, lambda free) Model 2

steps <-
  c(
    0.005,
    0.010,
    0.050,
    0.100,
    0.250,
    0.350,
    0.500,
    0.650,
    0.750,
    0.900,
    0.950,
    0.990,
    0.995,
    1
  )
nsteps <- length(steps)

pars2 <- c(mean(v) / 4, mean(d), 0.5)
weights <-
```

```
  c(1,
    0.99,
    0.95,
    0.90,
    0.80,
    0.75,
    0.60,
    0.60,
    0.75,
    0.80,
    0.90,
    0.95,
    0.99,
    1)

wt <- rep(1, number)
for (i in 1:number) {
  for (j in 2:nsteps) {
    if (-si[i] * qnorm(steps[j]) <= d[i] &&
        d[i] <= -si[i] * qnorm(steps[j - 1]))
      wt[i] = j
  }
  if (d[i] <= -si[i] * qnorm(steps[nsteps - 1]))
    wt[i] = nsteps
}

orig_lambda_vw2 <-
  optim(
    par = pars2,
    fn = neglike4,
    lower = c(0, -Inf, 0.01),
```

```
    method = "L-BFGS-B",
    hessian = TRUE
  )
orig_lambda_vw2_pars <- rbind(orig_lambda_vw2$par)
write.table(
  orig_lambda_vw2_pars,
  "orig_lambda_vw2.csv",
  append = TRUE,
  row.names = FALSE,
  col.names = FALSE,
  sep = ","
)

## Bayes lambda, weights "correctly" specified

steps <- c(.05, 1.0)
npred <- 0
intercept <- TRUE
XX <- matrix(nrow = length(dummy), ncol = 1)
XX[, 1] <- rep(1, length(dummy))
number <- length(dummy)
nsteps <- 2
effect <- d

si <- sqrt(v)

wt <- rep(1, number)
for (i in 1:number) {
  for (j in 2:nsteps) {
    if (-si[i] * qnorm(steps[j]) <= d[i] &&
        d[i] <= -si[i] * qnorm(steps[j - 1]))
```

```
    wt[i] = j
  }
  if (d[i] <= -si[i] * qnorm(steps[nsteps - 1]))
    wt[i] = nsteps
}

modelstring = "
model {
vcinv ~ dgamma(.001,.001)
vc <- 1/vcinv
mn ~ dnorm(0.2, 1.0E-5)
lambda ~ dunif(0, 100)

#w[1] ~ dunif(0.99999, 1.00001)
for(j in 1:nsteps){
w[j] ~ dunif(0, 1)
}

for(i in 1:number) {

for(j in 1:nsteps){
a1[i,j] <- ifelse((dummy[i]==0 && wt[i]==j),log(w[j]),0)
a2[i,j] <- ifelse((dummy[i]==1 && wt[i] > 1 && wt[i]==j && steps[j] > 0.05), log(lambda*w[j]),0)
a3[i,j] <- ifelse((dummy[i]==1 && wt[i]==j && steps[j] <= 0.05), log(w[j]),0)
a4[i,j] <- ifelse((dummy[i]==1 && wt[i]==j && wt[i]==1), log(w[j]),0)

bi[i,j] <- -sqrt(v[i])*qnorm(steps[j],0,1)
}

a[i] <- sum((a1[i,]+a2[i,]+a3[i,]+a4[i,]))
```

```
eta[i] <- sqrt(v[i] + vc)
bilast[i] <- -sqrt(v[i])*qnorm(steps[(nsteps-1)],0,1)

Bij[i,1] <- 1-pnorm( ((bi[i,1]-mn)/eta[i]), 0, 1 )

for(j in 2:(nsteps-1)){
Bij[i,j] <- pnorm( ((bi[i,(j - 1)]-mn)/eta[i]), 0, 1) - pnorm( ((bi[i,j]-mn)/eta[i]), 0, 1)
}

Bij[i,nsteps] <- pnorm( ((bilast[i]-mn)/eta[i]), 0, 1 )

for(j in 1:nsteps){
d1[i,j] <- ifelse((dummy[i]==1 && j > 1 && steps[j] > 0.05), (lambda*w[j]*Bij[i,j]), (w[j]*Bij[i,j]))
}

d[i] <- -log(sum(d1[i,]))

b[i] <- -1/2 * ((effect[i] - mn)/eta[i])^2
c[i] <- -log(eta[i])

L[i] <- a[i] + b[i] + c[i] + d[i] - 2000
dummy2[i] ~ dpois(-L[i])
}

}
"
writeLines(modelstring, con = "modelstring.bug")

dummy2 <- rep(0, length(d))

jags_params <- c("mn", "vcinv", "w", "lambda")
```

```
inits1 <-
  list(
    "mn" = 0.2,
    "vcinv" = 0.001,
    "w" = c(0.5, 0.5),
    "lambda" = 0.5
  )
inits2 <-
  list(
    "mn" = 0,
    "vcinv" = 0.001,
    "w" = c(0.2, 0.2),
    "lambda" = 1
  )
inits3 <-
  list(
    "mn" = 0.1,
    "vcinv" = 0.001,
    "w" = c(0.7, 0.7),
    "lambda" = 0.7
  )
jags.inits <- list(inits1, inits2, inits3)


jags.fit <- jags(
  data = list(
    'effect' = d,
    'v' = v,
    'wt' = wt,
    'number' = number,
```

```
    'dummy2' = dummy2,
    'dummy' = dummy,
    'steps' = steps,
    'nsteps' = nsteps
  ),
  parameters.to.save = jags_params,
  inits = jags.inits,
  n.chains = 3,
  n.burnin = 1000,
  n.iter = 5000,
  n.thin = 1,
  model.file = "modelstring.bug"
)

#update(jags, n.iter=30000, progress.bar="text")

# output <- coda.samples(jags.fit, n.iter = 10000)

# output <- coda.samples(jags, c("mn", "vcinv", "w", "lambda"), n.iter = 10000)

save(jags.fit,
     file = paste0("output ID ", round(
       as.numeric(Sys.time()) * (sample(1:100, 1, replace = TRUE))
     ), ".rda"),
     compress = "xz")

runtime <- toc()
runtime <- runtime$toc - runtime$tic

write.table(
  runtime,
```

```
            "runtime.csv",
            append = TRUE,
            row.names = FALSE,
            col.names = FALSE,
            sep = ","
          )

          start <- start + 1

        }
      }, warning = function(war) {
        print(paste("MY_WARNING: ", war))
        print(c("warning at", start))
        return(start)
      },
      error = function(err) {
        print(paste("MY_ERROR: ", err))
        print(c("restarted on", start))
        return(start + 1)
      },
      finally = {
      })
    }
    }

finish <- Sys.time()
```

Appendix C: R Code for Examples and Plots

```
future <- read.csv("K:/Dropbox/feeling-the-future.csv", header=TRUE)

future_g <- future$ES
future_v <- future$SE^2

install.packages("metafor")
install.packages("tidyverse")
install.packages("ggplot2")
install.packages("ggthemes")

library("metafor")
library("tidyverse")
library("ggplot2")
library("ggthemes")

early <- future %>% filter(Early == 1)
later <- future %>% filter(Early == 0)
effects <- c(early$ES, later$ES)
variances <- c((early$SE^2), (later$SE^2))

rma(later$ES, later$SE, method="ML")

plot(early$SE, early$ES)
plot(later$SE, later$ES)

## The following code generates Figures 1 and 2:
```

```
p_early <- ggplot(early, aes(x = SE, y = ES)) +
  geom_point(size=3) +
  scale_x_continuous(breaks = round(seq(min(early$SE), max(early$SE), by = 0.02),2)) +
  scale_y_continuous(breaks = round(seq(-0.4, 0.5, by = 0.05),1), limits=c(-0.4,0.5))+
  ggtitle("Studies Published Before 2011")+
  geom_hline(yintercept = 0.1141)

p_early + theme_base()


p_later <- ggplot(later, aes(x = SE, y = ES)) +
  geom_point(size=3) +
  scale_x_continuous(breaks = round(seq(min(later$SE), max(later$SE), by = 0.02),2)) +
  scale_y_continuous(breaks = round(seq(-0.4, 0.5, by = 0.05),1), limits=c(-0.4,0.5))+
  ggtitle("Studies Published After 2011")+
  geom_hline(yintercept = 0.0497)

p_later + theme_base()


##### This code yields the results of the adjusted and unadjusted lambda models presented in Chapter 4, along with the likelihood-ratio tests #####

weightfunct(effects, variances, lambda_model=dummy, steps=c(0.025, 0.05, 0.10, 0.50, 0.90, 1.00), fe=TRUE)

weightfunct(effects, variances, lambda_model=dummy, steps=c(0.025, 0.05, 0.10, 0.50, 0.90, 1.00), fe=TRUE)[2]

#Value (-116.7981) vs:

weightfunct(effects, variances, steps=c(0.025, 0.05, 0.10, 0.50, 0.90, 1.00), fe=TRUE)[2] #Value (-116.1549)
```

```
chisqdiff <- 116.7981-116.1549
1 - pchisq(chisqdiff, 1)

###
```

Appendix D: Extra Simulation Plots From Chapter 4

Figure D1 shows estimates of the mean in cells where I^2 is 25% and the bias pattern is "None vs. Weak."

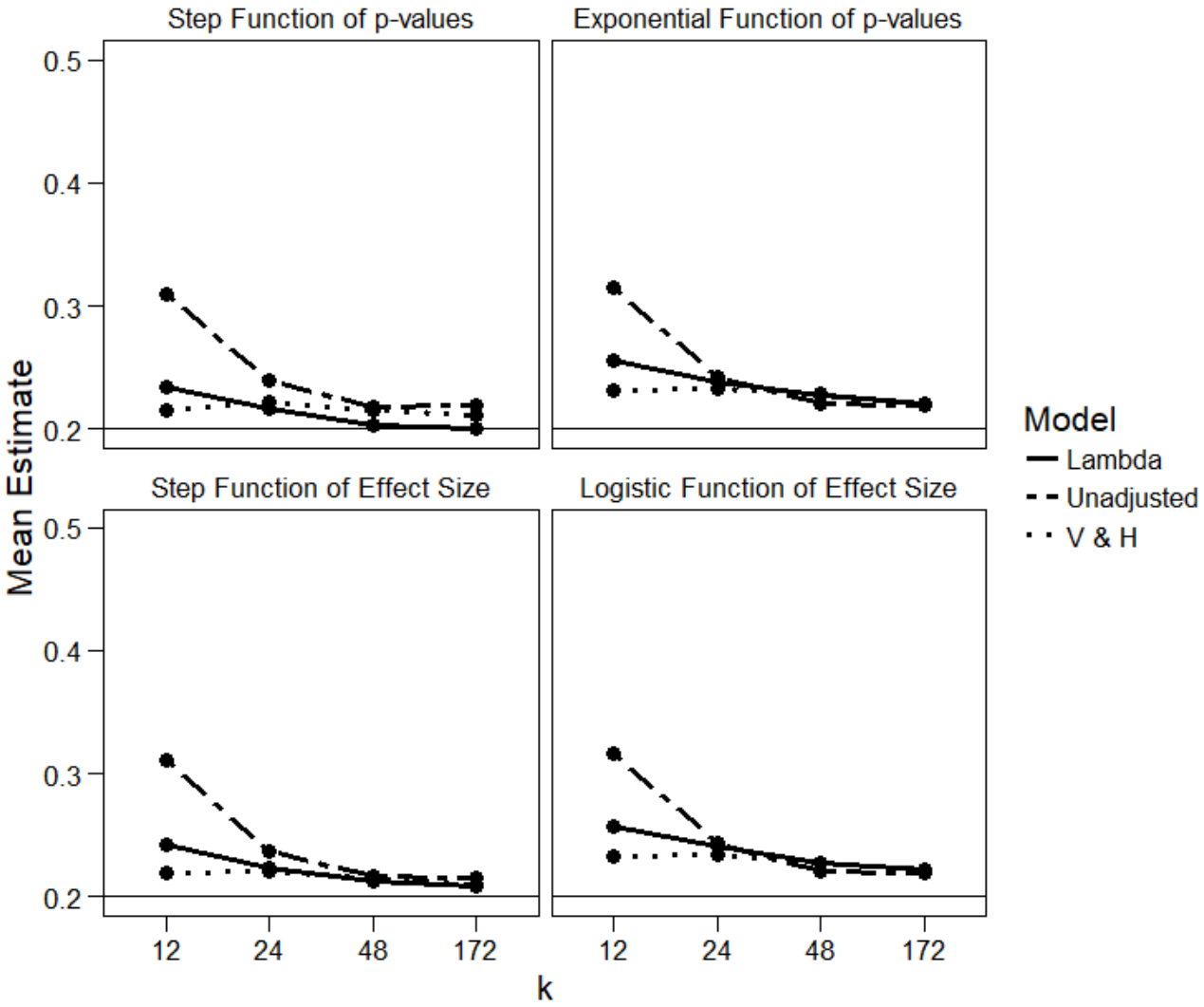


Figure D1.

Figure D2 shows estimates of the mean in cells where I^2 is 50% and the bias pattern is "None vs. Weak."

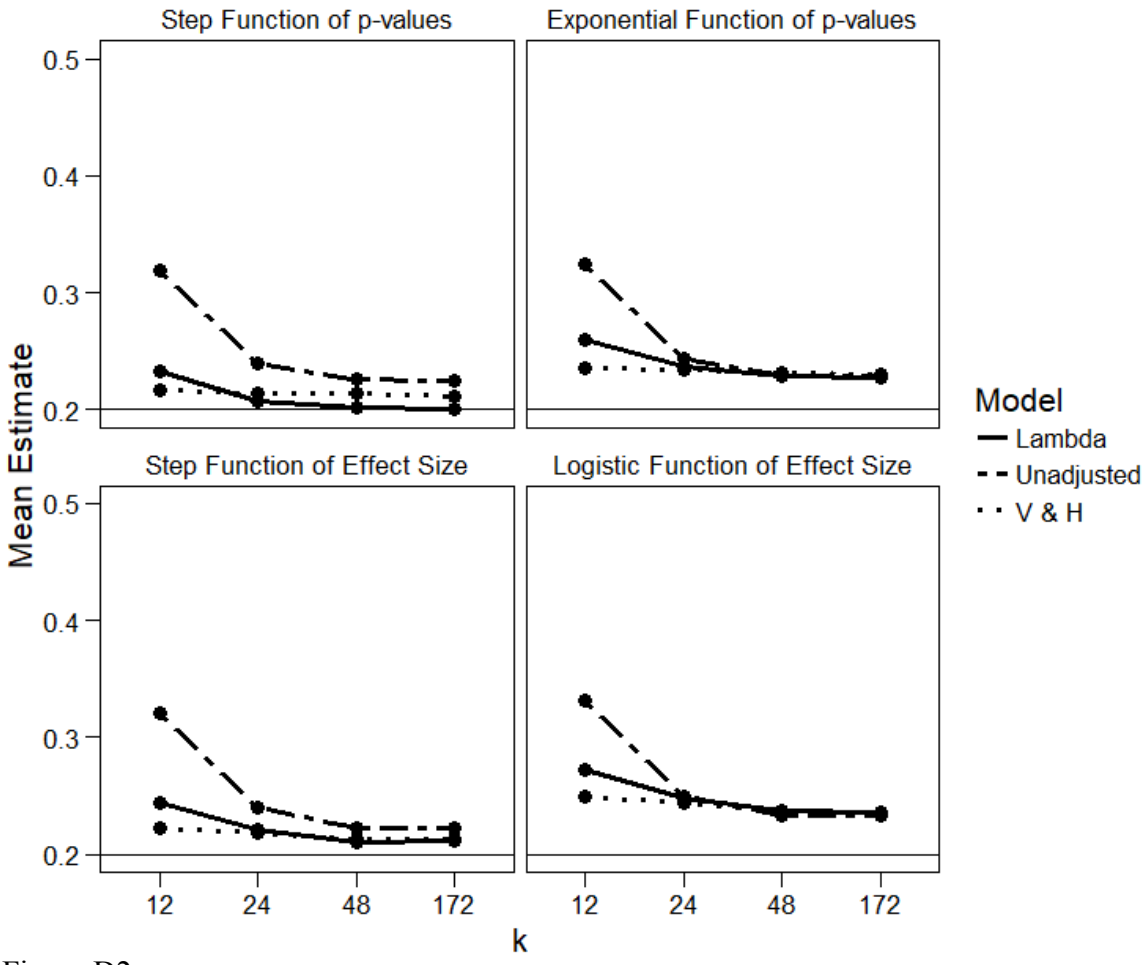


Figure D2.

Figure D3 features estimates of the mean in cells with I^2 of 25% and a "None vs. Strong" bias pattern:

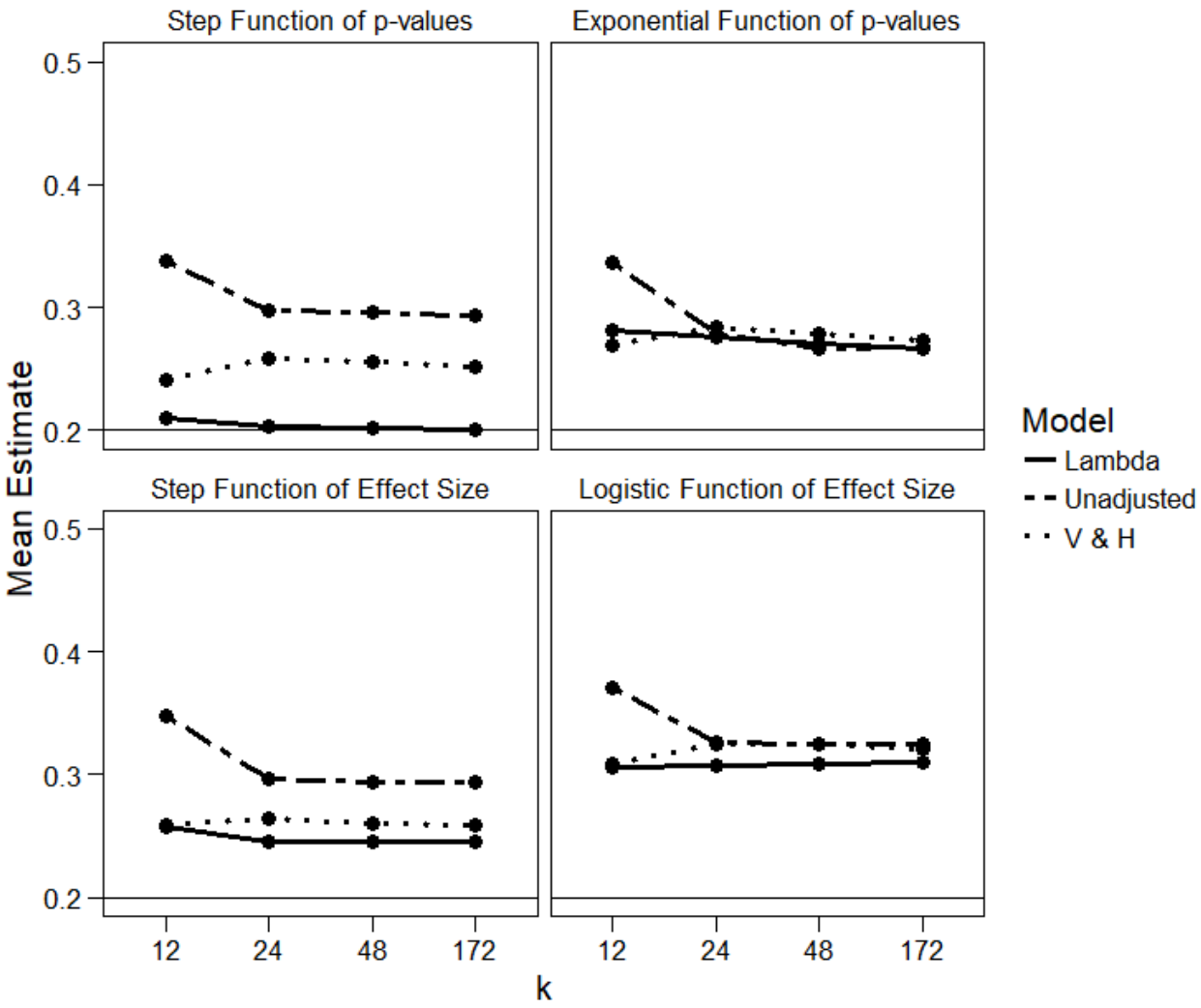


Figure D3.

Figure D4 features estimates of the mean in cells with I^2 of 50% and a "None vs. Strong" bias pattern:

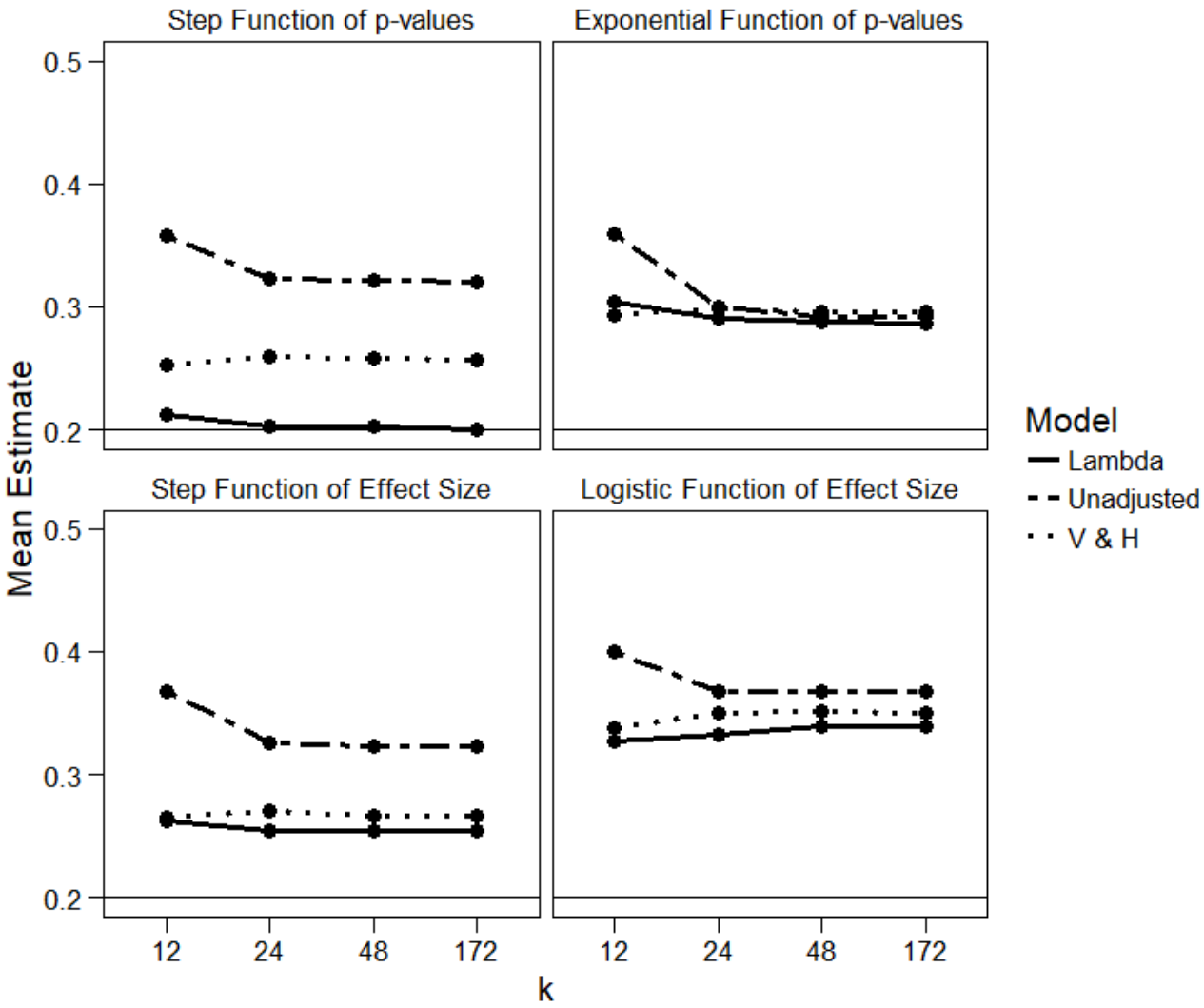


Figure D4.

Figure D5 features estimates of the mean in cells with I^2 of 25% and "Weak vs. Strong."

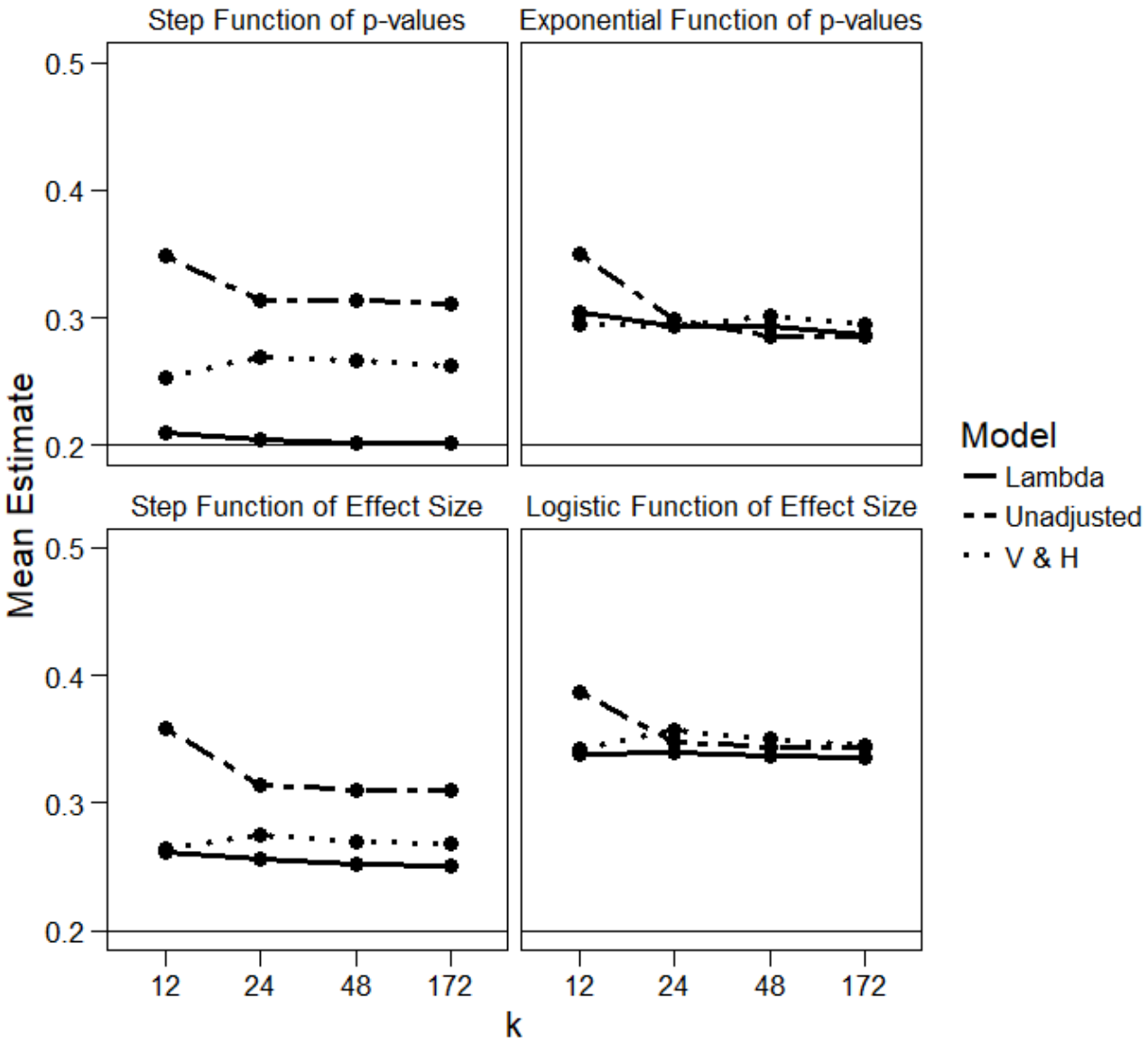


Figure D5.

Figure D6 features estimates of the mean in cells with I^2 of 50% and "Weak vs. Strong."

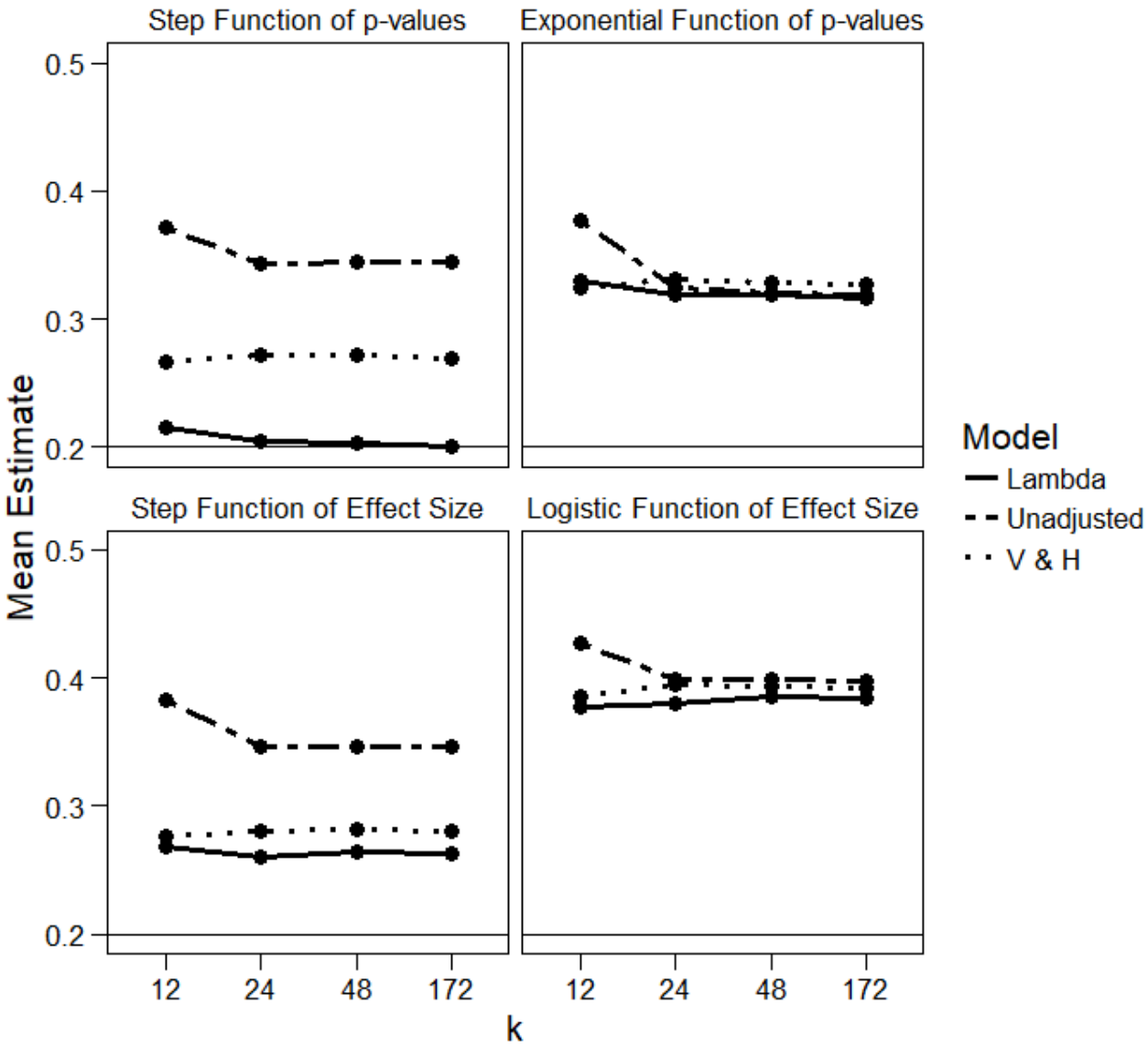
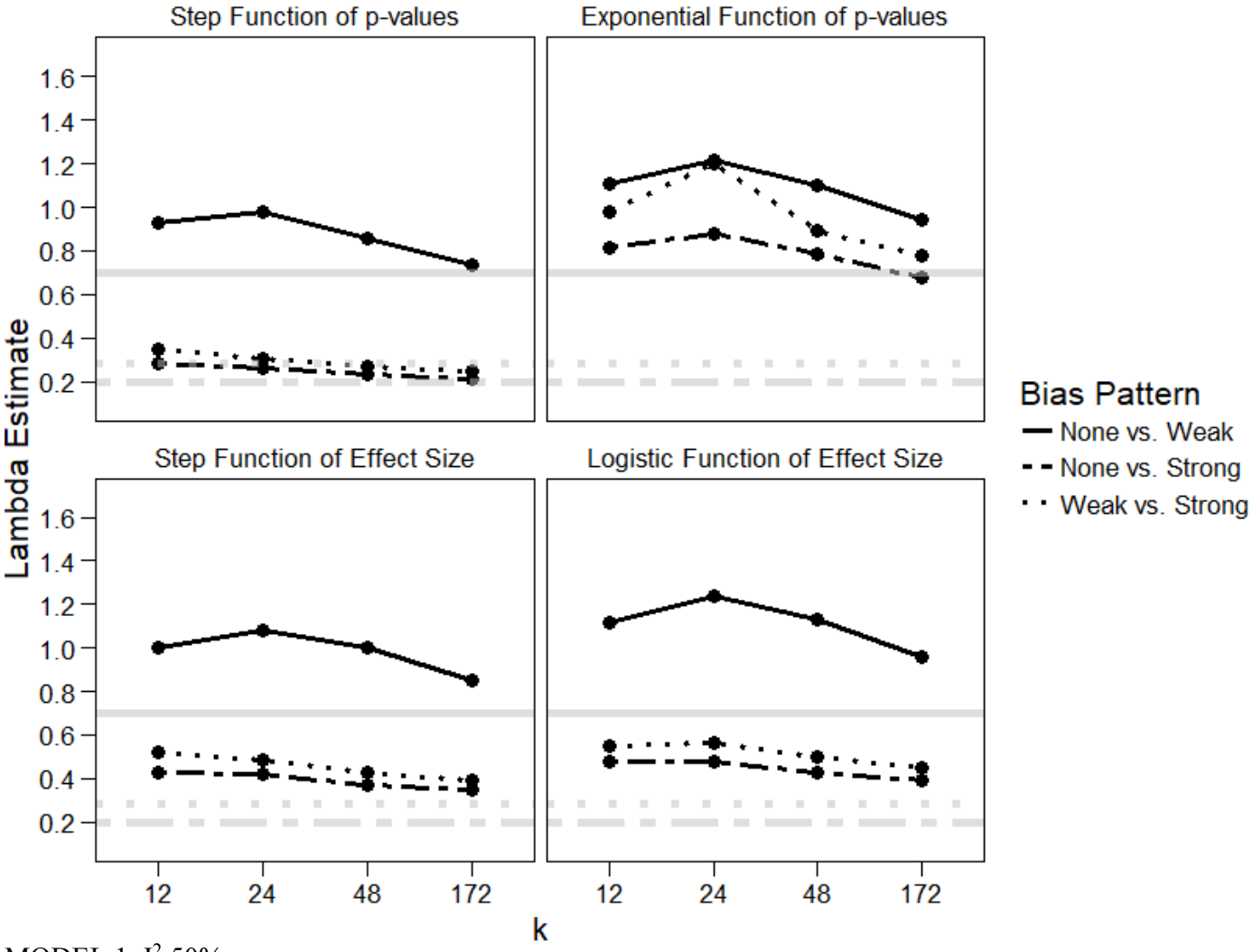

Step Function of p-values
Exponential Function of p-values
Step Function of Effect Size
Logistic Function of Effect Size
Mean Estimate
0.5
0.4
0.3
0.2
12
24
48
172
k
Model
Lambda
Unadjusted
V & H

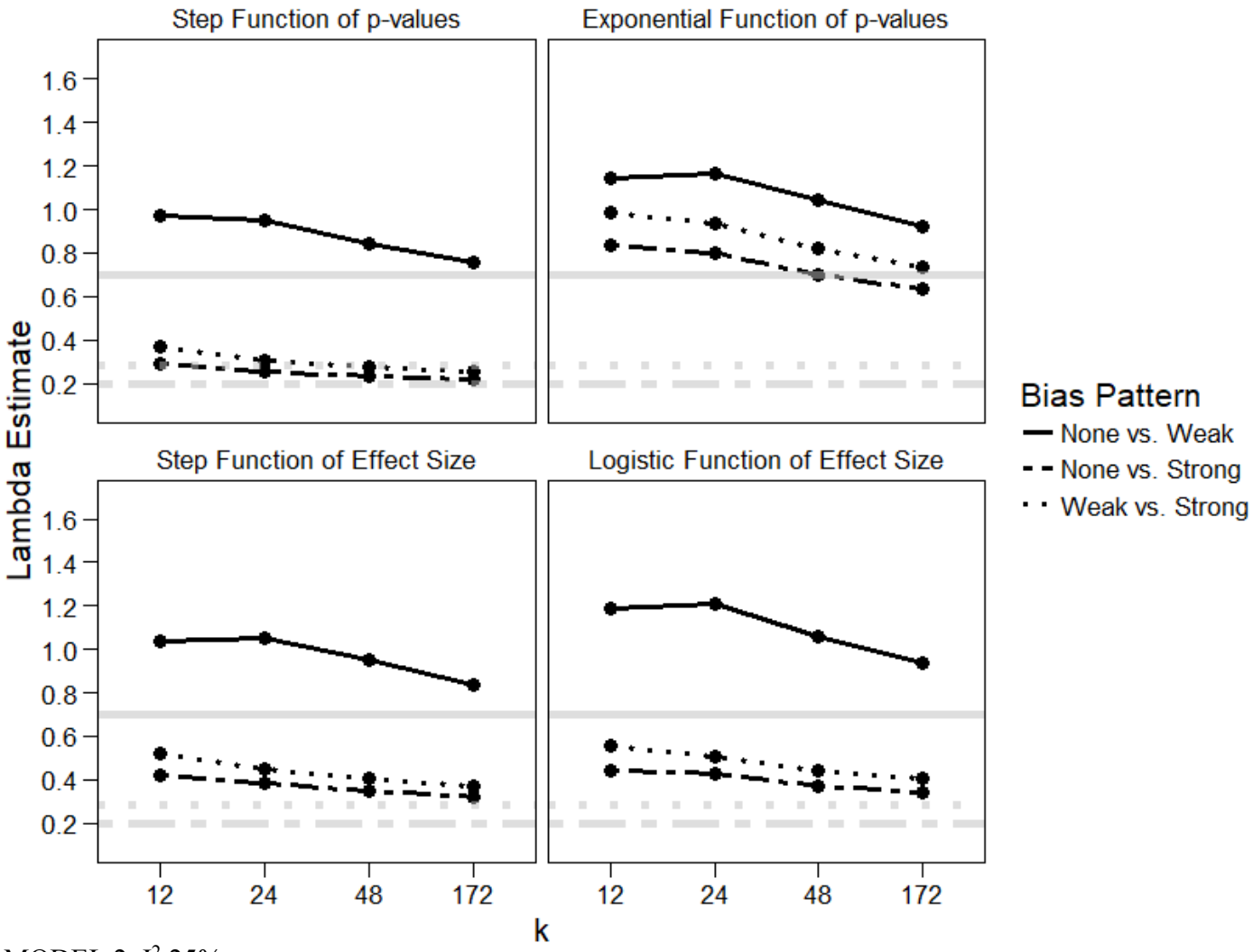
Figure D6.

Appendix E: Extra Simulation Plots From Chapter 5

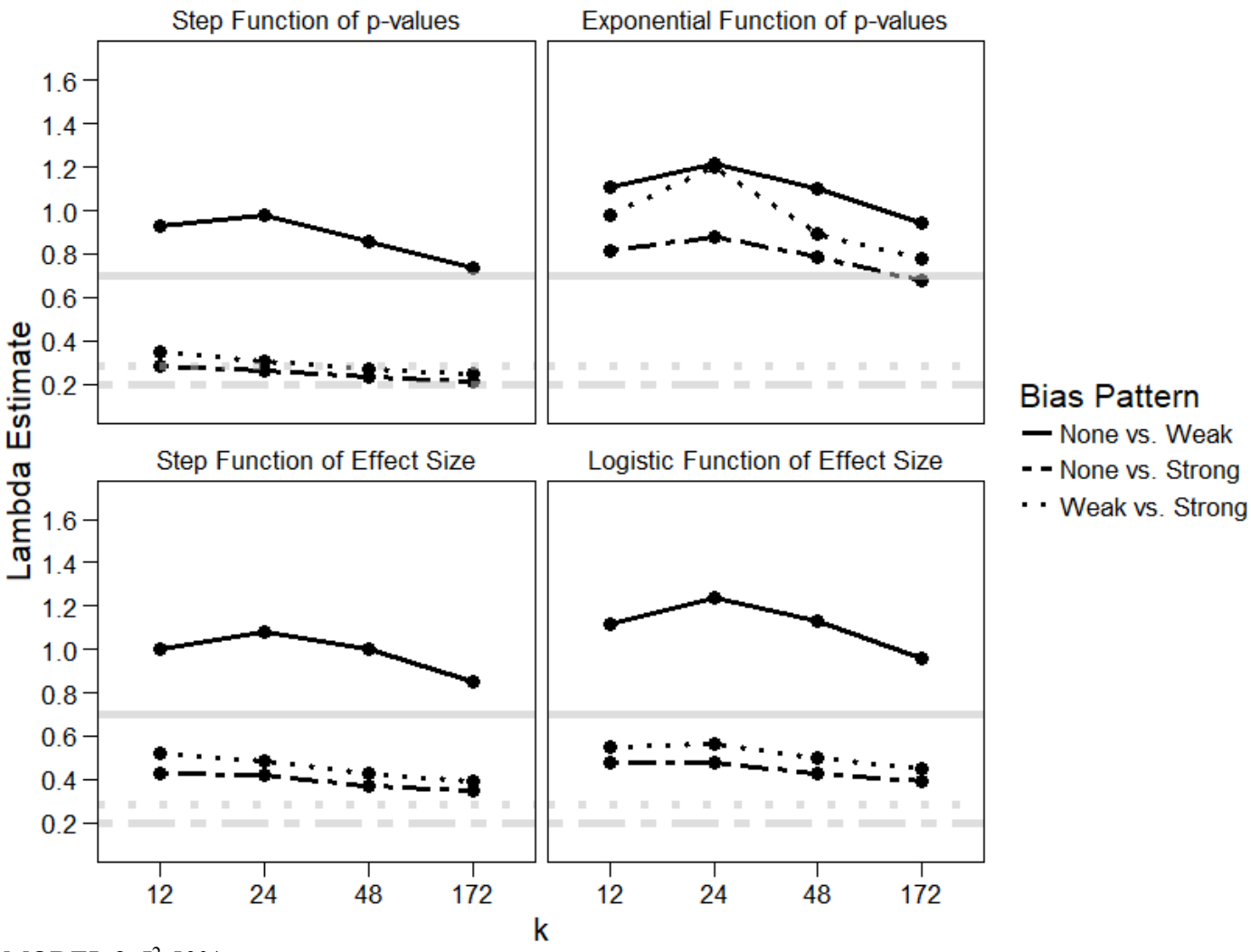
MODEL 1, I^2 25%



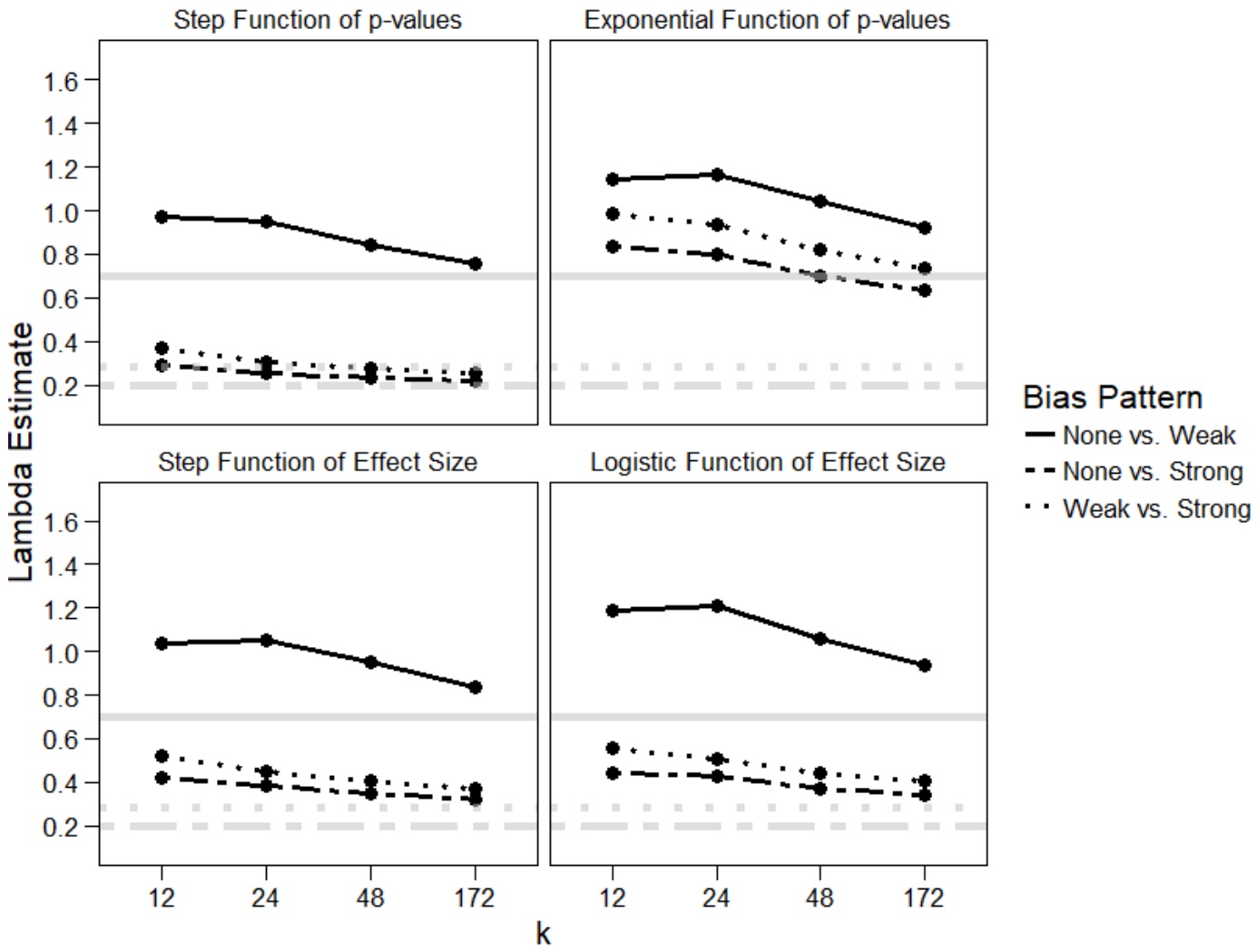
MODEL 1, I^2 50%


Step Function of p-values
Exponential Function of p-values
Step Function of Effect Size
Logistic Function of Effect Size
Lambda Estimate
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
12
24
48
172
k
Bias Pattern
None vs. Weak
None vs. Strong
Weak vs. Strong

MODEL 2, I^2 25%


Step Function of p-values
Exponential Function of p-values
Step Function of Effect Size
Logistic Function of Effect Size
Lambda Estimate
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
12
24
48
172
k
Bias Pattern
None vs. Weak
None vs. Strong
Weak vs. Strong

MODEL 2, I^2 50%


Step Function of p-values
Exponential Function of p-values
Step Function of Effect Size
Logistic Function of Effect Size
Lambda Estimate
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
12
24
48
172
k
Bias Pattern
None vs. Weak
None vs. Strong
Weak vs. Strong