Supplementary materials for "A penalization approach to random-effects meta-analysis"

APPENDIX A. SOME THEORETICAL RESULTS

A.1. Large tuning parameter in penalized likelihood of meta-analysis

The target function in the minimization problem of the penalized likelihood is

$$S(\tau^2) = \sum_{i=1}^{n} \left[\log(s_i^2 + \tau^2) + \frac{(y_i - \widehat{\mu}(\tau^2))^2}{s_i^2 + \tau^2} \right] + \lambda p(\tau^2).$$

Taking the derivative of $S(\tau^2)$ with respect to τ^2, we have

$$\begin{aligned} S'(\tau^2) &= \sum_{i=1}^{n} \frac{1}{s_i^2 + \tau^2} - \sum_{i=1}^{n} \frac{(y_i - \widehat{\mu}(\tau^2))^2}{(s_i^2 + \tau^2)^2} - 2\widehat{\mu}'(\tau^2) \sum_{i=1}^{n} \frac{y_i - \widehat{\mu}(\tau^2)}{s_i^2 + \tau^2} + \lambda p'(\tau^2) \\ &= \sum_{i=1}^{n} \frac{1}{s_i^2 + \tau^2} - \sum_{i=1}^{n} \frac{(y_i - \widehat{\mu}(\tau^2))^2}{(s_i^2 + \tau^2)^2} + \lambda p'(\tau^2) \\ &> \lambda p'(\tau^2) - \sum_{i=1}^{n} \frac{(y_i - \widehat{\mu}(\tau^2))^2}{(s_i^2 + \tau^2)^2}. \end{aligned}$$

A term is cancelled out in the above first step given $\widehat{\mu}(\tau^2) = \frac{\sum_{i=1}^{n} y_i/(s_i^2+\tau^2)}{\sum_{i=1}^{n} 1/(s_i^2+\tau^2)}$, thus

$$2\widehat{\mu}'(\tau^2) \sum_{i=1}^{n} \frac{y_i - \widehat{\mu}(\tau^2)}{s_i^2 + \tau^2} = 2\widehat{\mu}'(\tau^2) \left(\sum_{i=1}^{n} \frac{y_i}{s_i^2 + \tau^2} - \widehat{\mu}(\tau^2) \sum_{i=1}^{n} \frac{1}{s_i^2 + \tau^2} \right) = 0.$$

If $p(\tau^2) = \tau^2$, then $p'(\tau^2) = 1$. Note that, for sufficiently large τ^2, $\sum_{i=1}^{n} \frac{(y_i - \widehat{\mu}(\tau^2))^2}{(s_i^2+\tau^2)^2}$ decreases toward 0 as τ^2 increases because the denominator increases toward $+\infty$. Additionally, $\widehat{\mu}(\tau^2)$ is bounded (converging to the arithmetic mean of y_i's as τ^2 increases), implying that the numerator is bounded. Consequently, $\sum_{i=1}^{n} \frac{(y_i - \widehat{\mu}(\tau^2))^2}{(s_i^2+\tau^2)^2}$ has an upper bound, denoted by $\lambda_{\max}$, and thus

$$S'(\tau^2) > \lambda - \lambda_{\max}.$$

When $\lambda > \lambda_{\max}$, $S'(\tau^2) > 0$ for any value of τ^2, so the target function $S(\tau^2)$ takes the minimum at $\widehat{\tau}^2(\lambda) = 0$. Therefore, candidate values for the tuning parameter λ can be effectively selected from $[0, \lambda_{\max}]$.

A.2. The loss function for tuning the between-study standard deviation

When tuning the between-study standard deviation τ_{t}, recall that the loss function is

$$\widehat{L}(\tau_{\mathrm{t}}) = \left\{ \frac{1}{n} \sum_{i=1}^{n} \frac{(y_i - \widehat{\mu}_{(-i)}(\tau_{\mathrm{t}}))^2}{s_i^2 + \widehat{\tau}^2_{\mathrm{RE}(-i)} + \frac{\sum_{j \neq i} (s_j^2 + \widehat{\tau}^2_{\mathrm{RE}(-i)})/(s_j^2 + \tau_{\mathrm{t}}^2)^2}{\left[\sum_{j \neq i} 1/(s_j^2 + \tau_{\mathrm{t}}^2)\right]^2}} \right\}^{\frac{1}{2}}.$$

As $\tau_{\mathrm{t}} \to +\infty$, we have $\widehat{\mu}_{(-i)}(\tau_{\mathrm{t}}) \to \frac{1}{n-1} \sum_{j \neq i} y_j$, so the numerator of the loss function is bounded. In addition, $\widehat{\tau}^2_{\mathrm{RE}(-i)}$ depends only on the dataset (i.e., y_j and s_j^2) and thus is unrelated to τ_{t}. Suppose that for $i \neq j = 1, \ldots, n$,

$$0 \leq \widehat{\tau}^2_{\mathrm{RE}(-i)} \leq C_1 \text{ and } 0 < C_3 \leq s_j^2 \leq C_2,$$

where C_1, C_2, and C_3 are constants which are independent of τ_t. For the denominator in the loss function, we have

$$\sum_{j\neq i}\frac{s_j^2+\widehat{\tau}^2_{\mathrm{RE}(-i)}}{(s_j^2+\tau_t^2)^2}\leq\sum_{j\neq i}\frac{C_1+C_2}{(C_3+\tau_t^2)^2}=\frac{(n-1)(C_1+C_2)}{(C_3+\tau_t^2)^2};$$
$$\left(\sum_{j\neq i}\frac{1}{s_j^2+\tau_t^2}\right)^2\geq\left(\sum_{j\neq i}\frac{1}{C_2+\tau_t^2}\right)^2=\frac{(n-1)^2}{(C_2+\tau_t^2)^2}.$$

Consequently,

$$\frac{\sum_{j\neq i}(s_j^2+\widehat{\tau}^2_{\mathrm{RE}(-i)})/(s_j^2+\tau_t^2)^2}{\left[\sum_{j\neq i}1/(s_j^2+\tau_t^2)\right]^2}\leq\frac{(C_1+C_2)(C_2+\tau_t^2)^2}{(n-1)(C_3+\tau_t^2)^2}.$$

When $\tau_t\to+\infty$, the right side in the above inequality converges to $\frac{C_1+C_2}{n-1}$. Thus, the denominator of the loss function has an upper bound of $C_1+C_2+\frac{C_1+C_2}{n-1}$. Therefore, unlike using the tuned value τ_t^2 to estimate τ^2 in the loss function, the loss function with the random-effects estimate $\widehat{\tau}^2_{\mathrm{RE}(-i)}$ is bounded and does not have a trivial trend toward 0 as τ_t increases.

A.3. The comparison of the standard error of the estimated overall effect size between the proposed penalization method and the random-effects model

For a given meta-analysis, the standard error of the overall effect size estimate under the RE model is

$$\mathrm{SE}(\widehat{\mu}_{\mathrm{RE}})=\sqrt{\frac{1}{\sum_{i=1}^n w_i}},$$

and that of the penalization method is

$$\mathrm{SE}(\widehat{\mu}_{\mathrm{PRE}})=\sqrt{\frac{\sum_{i=1}^n\frac{(w_i')^2}{w_i}}{(\sum_{i=1}^n w_i')^2}},$$

where $w_i=1/(s_i^2+\widehat{\tau}^2_{\mathrm{RE}})$ and $w_i'=1/(s_i^2+\tau^2_{\mathrm{PRE}})$. Note that τ^2_{PRE} is the between-study variance estimate of the penalization method. It is possible for $\tau^2_{\mathrm{PRE}}=\widehat{\tau}^2$ when the tuning parameter is λ or $\tau^2_{\mathrm{PRE}}=\tau_t^2$ when the tuning parameter is τ. We take the ratio of standard errors to compare their magnitudes:

$$\begin{aligned}\frac{\mathrm{SE}(\widehat{\mu}_{\mathrm{PRE}})}{\mathrm{SE}(\widehat{\mu}_{\mathrm{RE}})}&=\frac{\frac{1}{\sum w_i'}\sqrt{\sum\frac{(w_i')^2}{w_i}}}{\sqrt{\frac{1}{\sum w_i}}}\\&=\sqrt{\sum\frac{w_i}{(\sum w_i')^2}\sum\frac{(w_i')^2}{w_i}}\\&\geq\sqrt{\sum\frac{\sqrt{w_i}}{\sum w_i'}\cdot\frac{w_i'}{\sqrt{w_i}}}=1.\end{aligned}$$

The above comparative relationship is obtained by using the Cauchy–Schwarz inequality. Therefore, at the same statistical significance level, the confidence interval of the overall effect size estimate of the penalization method may be wider than that of the RE model.

B. ADDITIONAL SIMULATION RESULTS

In addition to the simulation study in the main content, we performed simulation studies under a broader range of settings. Specifically, the simulated meta-analyses in Table S1 used similar parameters to the simulation study in the main content, but it focused on the settings without outliers. The simulated meta-analyses had low to moderate heterogeneity, with I^2 approximately equal to 15%, 25%, and 50%. The results indicate that, despite the absence of outliers, the penalization methods still produced 95% confidence intervals with better coverage probabilities, although the RE model generally produced smaller MSEs. This was not surprising because the simulation settings favored the RE model.

TABLE S1 Biases, mean squared errors (MSEs), and 95% confidence interval coverage probabilities (CPs) in percentage for simulated meta-analyses with low to moderate heterogeneity using the four methods: common-effect model (CE); random-effects model (RE); penalization method for the random-effects model by tuning λ (PRE-λ); and penalization method for the random-effects model by tuning the between-study standard deviation (PRE-τ).

No. of	CE			RE			PRE-λ			PRE-τ		
studies	Bias	MSE	CP	Bias	MSE	CP	Bias	MSE	CP	Bias	MSE	CP
$I^2 \approx 15\%$:												
5	0.005	0.054	83.5	0.005	0.054	88.0	0.005	0.054	88.1	0.005	0.054	88.0
10	0.004	0.027	80.6	0.003	0.025	88.9	0.003	0.026	88.8	0.004	0.025	88.9
25	0.003	0.011	76.4	0.002	0.010	89.8	0.002	0.010	90.2	0.002	0.010	90.0
$I^2 \approx 25\%$:												
5	0.005	0.074	75.4	0.005	0.069	84.8	0.005	0.070	84.8	0.005	0.070	84.8
10	0.005	0.041	71.9	0.004	0.033	87.3	0.004	0.034	87.2	0.004	0.034	87.3
25	0.004	0.018	65.6	0.002	0.013	90.1	0.003	0.014	90.7	0.002	0.014	90.9
$I^2 \approx 50\%$:												
5	0.005	0.153	59.0	0.005	0.121	80.4	0.004	0.129	80.5	0.003	0.125	80.4
10	0.008	0.094	52.8	0.005	0.057	86.9	0.005	0.064	87.5	0.004	0.063	87.4
25	0.006	0.044	46.8	0.003	0.023	92.1	0.004	0.028	93.9	0.003	0.027	93.8

Table S2 presents the results of simulated meta-analyses with random effects generated from the scaled t_5 distribution. It focused on the case of n=30. The different scale parameters for the random effects led to different between-study variances, with I^2 approximately equal to 0%, 20%, 50%, and 80%. The penalization methods generally produced 95% confidence intervals with better coverage probabilities. The RE model produced smaller MSEs when the scale parameter was relatively large.

TABLE S2 Biases, mean squared errors (MSEs), and 95% confidence interval coverage probabilities (CPs) in percentage for simulated meta-analyses with scaled t_5-distributed random effects (n=30) using the four methods: common-effect model (CE); random-effects model (RE); penalization method for the random-effects model by tuning λ (PRE-λ); and penalization method for the random-effects model by tuning the between-study standard deviation (PRE-τ).

Scale	CE			RE			PRE-λ			PRE-τ		
parameter	Bias	MSE	CP	Bias	MSE	CP	Bias	MSE	CP	Bias	MSE	CP
0	0.000	0.004	94.9	0.000	0.004	95.6	0.000	0.004	95.7	0.000	0.004	95.6
0.21	0.001	0.012	73.7	−0.001	0.009	92.1	−0.001	0.009	92.6	−0.001	0.009	92.5
0.43	0.002	0.039	48.1	−0.002	0.018	93.4	−0.001	0.022	95.2	−0.001	0.022	95.1
0.85	0.003	0.140	27.6	−0.003	0.050	93.7	−0.001	0.068	96.8	−0.002	0.066	97.1

All foregoing results of simulation studies were based on the maximum likelihood estimation. Tables S3–S5 used the restricted maximum likelihood (REML) estimate for τ^2 in Section 3.5 in the main content to calculate $\widehat{\mathrm{Var}}(\widehat{\mu})$. They correspond to Table 1

in the main context and Tables S1 and S2. The performance of the four methods had very slight changes when using the two different estimation methods.

Of note, although Table S3 and Table 1 in the main context focused on the same simulation settings, they might be based on different sets of simulated meta-analyses, because the REML estimation for τ^2 could not converge in some simulated meta-analyses, which were ignored in Table S3. More specifically, when $\tau = 0$ or 1 and the number of outliers was 0 or 1, the REML estimation of τ^2 failed to converge in some simulated meta-analyses. Under other simulation scenarios, all simulated meta-analyses could generate REML estimates for τ^2.

TABLE S3 Biases, mean squared errors (MSEs), and 95% confidence interval coverage probabilities (CPs) in percentage for simulated meta-analyses using the four methods based on the restricted maximum likelihood estimation: common-effect model (CE); random-effects model (RE); penalization method for the random-effects model by tuning λ (PRE-λ); and penalization method for the random-effects model by tuning the between-study standard deviation (PRE-τ).

No. of	CE			RE			PRE-λ			PRE-τ		
outliers	Bias	MSE	CP	Bias	MSE	CP	Bias	MSE	CP	Bias	MSE	CP
$\tau = 0$:												
0	0.001	0.004	95.4	0.001	0.004	96.2	0.001	0.004	96.3	0.000	0.004	96.2
1	0.098	0.036	71.8	0.080	0.015	95.9	0.061	0.011	97.6	0.076	0.014	96.4
2	0.202	0.090	46.3	0.184	0.045	91.8	0.145	0.035	96.4	0.173	0.044	93.4
3	0.300	0.161	27.1	0.291	0.097	79.2	0.238	0.079	91.1	0.264	0.088	86.9
4	0.400	0.249	13.0	0.397	0.169	58.0	0.339	0.143	82.4	0.356	0.151	78.7
5	0.498	0.356	6.1	0.499	0.260	32.1	0.443	0.230	71.1	0.451	0.234	68.9
6	0.597	0.481	2.4	0.600	0.371	12.9	0.551	0.342	60.7	0.555	0.344	60.1
$\tau = 0.5$:												
0	−0.003	0.033	48.9	−0.003	0.017	93.6	−0.002	0.020	95.1	−0.003	0.020	95.3
1	0.107	0.073	40.1	0.103	0.029	95.1	0.101	0.039	96.9	0.104	0.039	97.3
2	0.224	0.144	29.5	0.216	0.066	90.5	0.198	0.073	94.9	0.201	0.073	95.1
3	0.333	0.228	21.0	0.328	0.128	80.6	0.297	0.127	90.9	0.299	0.127	91.2
4	0.439	0.332	14.8	0.441	0.215	63.3	0.402	0.206	83.7	0.403	0.206	84.2
5	0.553	0.466	9.6	0.554	0.327	42.5	0.515	0.316	74.3	0.515	0.315	75.1
6	0.665	0.623	6.0	0.666	0.464	22.9	0.626	0.448	65.2	0.625	0.446	66.3
$\tau = 1$:												
0	−0.004	0.119	27.0	−0.002	0.044	94.1	−0.001	0.059	97.0	−0.002	0.058	97.1
1	0.135	0.184	23.7	0.138	0.064	94.3	0.140	0.094	97.2	0.137	0.091	97.5
2	0.284	0.299	19.0	0.280	0.123	89.7	0.268	0.154	95.3	0.264	0.150	95.7
3	0.421	0.434	15.0	0.421	0.222	81.5	0.396	0.246	91.4	0.393	0.242	92.0
4	0.556	0.600	11.7	0.562	0.362	67.3	0.531	0.380	85.8	0.527	0.374	86.6
5	0.699	0.813	8.7	0.704	0.541	50.1	0.670	0.555	78.2	0.668	0.550	79.2
6	0.841	1.064	6.0	0.846	0.761	31.9	0.811	0.770	69.5	0.809	0.763	70.6
$\tau = 2$:												
0	−0.005	0.467	14.6	0.000	0.145	94.2	−0.002	0.208	97.4	−0.003	0.207	97.4
1	0.214	0.630	12.7	0.223	0.196	94.2	0.219	0.294	97.6	0.215	0.290	97.8
2	0.449	0.917	9.5	0.447	0.346	89.6	0.436	0.458	95.6	0.430	0.449	95.9
3	0.665	1.254	8.3	0.670	0.595	82.7	0.641	0.699	92.6	0.636	0.691	93.1
4	0.879	1.669	6.9	0.894	0.945	70.5	0.857	1.042	87.6	0.851	1.028	88.3
5	1.105	2.204	5.4	1.118	1.395	55.6	1.079	1.490	80.5	1.072	1.469	81.6
6	1.330	2.831	4.2	1.341	1.945	37.8	1.304	2.037	72.9	1.299	2.021	73.7

TABLE S4 Biases, mean squared errors (MSEs), and 95% confidence interval coverage probabilities (CPs) in percentage for simulated meta-analyses with low to moderate heterogeneity using the four methods based on the restricted maximum likelihood estimation: common-effect model (CE); random-effects model (RE); penalization method for the random-effects model by tuning λ (PRE-λ); and penalization method for the random-effects model by tuning the between-study standard deviation (PRE-τ).

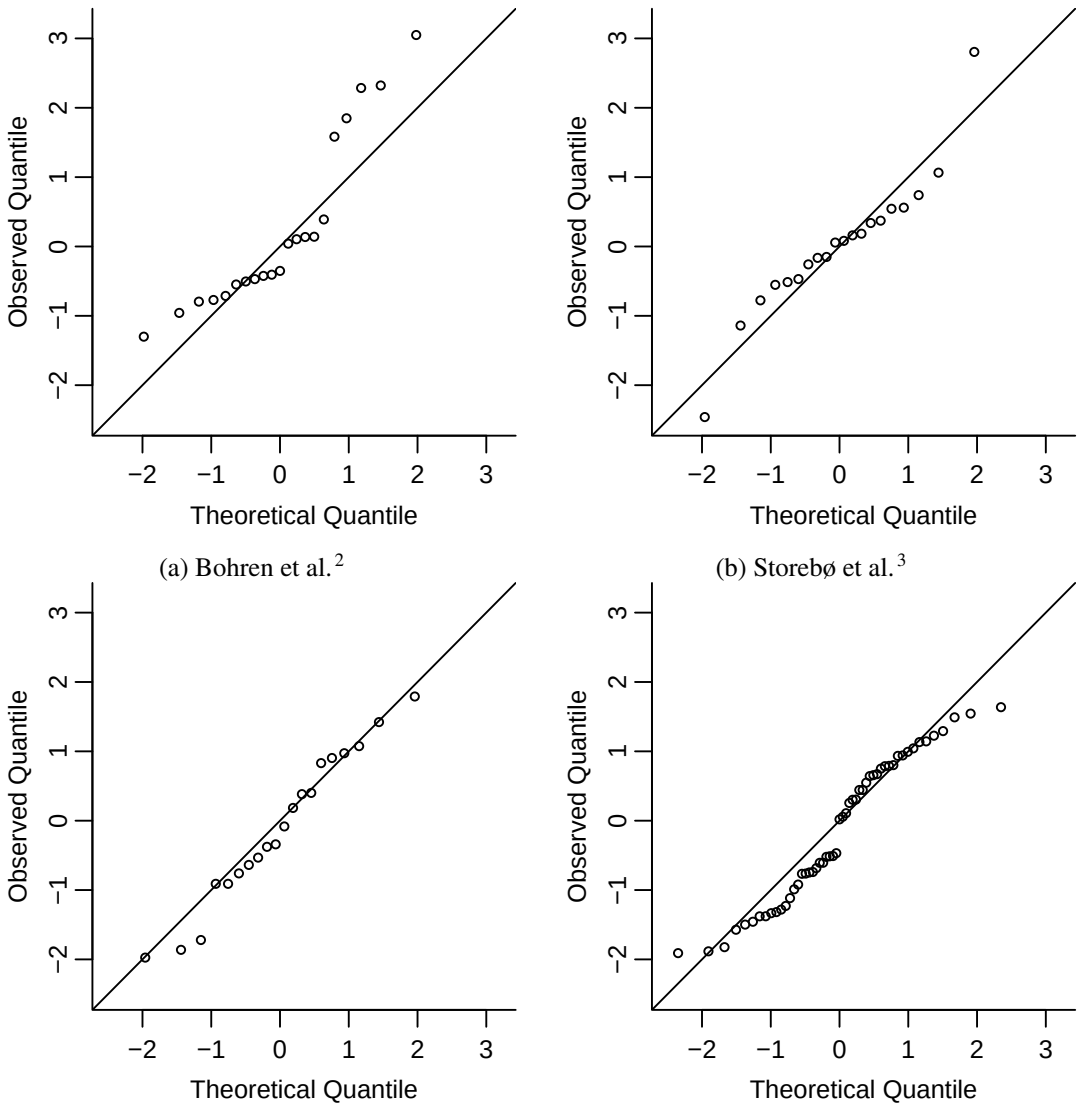
No. of	CE			RE			PRE-λ			PRE-τ		
studies	Bias	MSE	CP	Bias	MSE	CP	Bias	MSE	CP	Bias	MSE	CP
$I^2 \approx 15\%$:												
5	0.006	0.056	82.4	0.002	0.058	88.4	0.004	0.056	88.6	0.003	0.057	88.5
10	0.003	0.028	78.7	0.003	0.026	88.9	0.003	0.026	89.3	0.003	0.026	89.1
25	−0.000	0.012	76.8	0.000	0.010	91.4	−0.000	0.010	91.8	0.000	0.010	91.7
$I^2 \approx 25\%$:												
5	0.008	0.077	74.8	0.003	0.073	85.8	0.005	0.073	85.9	0.003	0.072	85.9
10	0.004	0.042	69.8	0.003	0.034	88.0	0.004	0.036	88.3	0.004	0.035	88.3
25	0.000	0.018	66.5	−0.000	0.013	91.3	−0.000	0.014	92.5	−0.000	0.014	92.2
$I^2 \approx 50\%$:												
5	0.013	0.160	58.2	0.004	0.124	83.7	0.008	0.134	83.9	0.006	0.126	83.8
10	0.007	0.097	52.0	0.005	0.059	88.6	0.003	0.068	89.2	0.002	0.066	89.2
25	0.000	0.046	45.1	0.000	0.023	92.2	−0.000	0.028	94.8	−0.000	0.028	94.5

TABLE S5 Biases, mean squared errors (MSEs), and 95% confidence interval coverage probabilities (CPs) in percentage for simulated meta-analyses with scaled t_5-distributed random effects (n=30) using the four methods based on the restricted maximum likelihood estimation: common-effect model (CE); random-effects model (RE); penalization method for the random-effects model by tuning λ (PRE-λ); and penalization method for the random-effects model by tuning the between-study standard deviation (PRE-τ).

Scale	CE			RE			PRE-λ			PRE-τ		
parameter	Bias	MSE	CP	Bias	MSE	CP	Bias	MSE	CP	Bias	MSE	CP
0	−0.001	0.004	94.9	−0.001	0.004	96.0	−0.001	0.004	96.0	−0.001	0.004	96.0
0.21	0.001	0.012	71.6	0.000	0.009	92.6	0.001	0.009	92.9	0.001	0.009	92.9
0.43	0.002	0.040	47.5	0.000	0.019	93.8	0.001	0.023	95.4	0.001	0.023	95.3
0.85	0.005	0.145	27.1	−0.001	0.050	94.5	0.002	0.069	96.8	0.002	0.069	96.9

C. EXAMINING THE BETWEEN-STUDY NORMALITY ASSUMPTION FOR THE FOUR EXAMPLES OF META-ANALYSES

We used the quantile–quantile (Q–Q) plots and the Shapiro–Wilk test of standardized effect estimates described by Wang and Lee [1] to examine the between-study normality assumption for the four examples of meta-analyses. The test's p-values were obtained, and the statistical significance level for normality was set to 0.1. For the meta-analysis by Bohren et al.,[2] the p-value is 0.004, which is far less than 0.1 (rejecting the null hypothesis). Its Q–Q plot is in Figure S1(a), which indicates that the distribution is skewed to the left. For the meta-analysis by Storebø et al.,[3] the p-value is 0.087, slightly less than 0.1. The Q–Q plot is in Figure S1(b), which seems to indicate that its distribution has thinner tails than the normal distribution. For the meta-analysis by Carless et al.,[4] the p-value is 0.704, which is much large than 0.1 (not rejecting the null hypothesis). Figure S1(c) presents its Q–Q plot, where the points are mostly in a straight line. For the meta-analysis by Bjelakovic et al.,[5] the p-value is 0.014, which is much less than 0.1 (rejecting the null hypothesis). Figure S1(d) gives its Q–Q plot, indicating that its distribution has thicker tails than the normal distribution.



(a) Bohren et al.[2]

(b) Storebø et al.[3]

(c) Carless et al.[4]

(d) Bjelakovic et al.[5]

FIGURE S1 Normal quantile–quantile plots for the four examples of meta-analyses.

APPENDIX D. TWO ADDITIONAL EXAMPLES

Two additional examples of meta-analyses were used to demonstrate that different conclusions could be obtained when tuning different parameters in the penalization methods. The first meta-analysis, consisting of 27 studies, was reported by Keus et al.[6] to compare the beneficial and harmful effects of laparoscopic versus open cholecystectomy in the subgroup of patients with symptomatic cholecystolithiasis. The second meta-analysis, consisting of 32 studies, was conducted by Adams et al.[7] to synthesize the effects of fluvastatin 20 mg/day on high-density lipoprotein. The effect measure in the first meta-analysis was (log) odds ratio, whereas that in the second meta-analysis was the mean difference. Forest plots of these two meta-analyses are presented in Figures S2 and S3.

For the meta-analysis by Keus et al.,[6] the CE model yielded an estimated overall OR of 0.68 with 95% CI (0.44, 1.06). In contrast, the RE model estimated the overall OR as 0.54 with 95% CI (0.30, 0.98) and $\widehat{\tau}_{\text{RE}} = 0.71$. The I^2 statistic was 13% with 95% CI (0%, 51%), and the p-value of the Q test was 0.27. Both results indicate a small degree of between-study heterogeneity. The minimum loss of the penalization method by tuning τ was achieved at $\tau_{\text{t}} = 0$, with the overall OR estimated as 0.68 with 95% CI (0.34, 1.35). However, the penalization method by tuning λ favored neither the CE nor RE model. It estimated the overall OR as 0.57 with 95% CI (0.31, 1.04) and $\widehat{\tau} = 0.56$. The vertical line at $\widehat{\tau} = 0$ in Figure S4(a) demonstrates that all $\lambda > \lambda_{\text{max}}$ led to zero estimates of τ, but they could correspond to different loss values.

For the meta-analysis by Adams et al.,[7] the CE model estimated the overall mean difference as 4.39 with 95% CI (4.03, 4.76); the RE model estimated it as 5.18 with 95% CI (4.47, 5.90) and $\widehat{\tau}_{\text{RE}} = 1.20$. The penalization method by tuning λ achieved the minimum loss at $\widehat{\tau} = \widehat{\tau}_{\text{RE}}$. The respective estimated overall mean difference was identical to that of the RE model. Nevertheless, the minimum loss of the penalization method by tuning τ was achieved at $\tau_{\text{t}} = 0.59$, less than $\widehat{\tau}_{\text{RE}}$, and the overall mean difference was estimated as 4.80 with 95% CI (4.03, 5.57). Figure S4(f) shows that studies 29 and 30 were outliers (i.e., absolute standardized residuals were greater than 3 under both the CE and RE settings). Moreover, if the two outlying studies were removed from the meta-analysis, the I^2 statistic reduced from 65% with 95% CI (44%, 95%) to 11% with 95% CI (0%, 71%). For this meta-analysis, the penalization method of tuning λ failed to penalize the overestimated heterogeneity, but the penalization method of tuning τ succeeded in this respect.

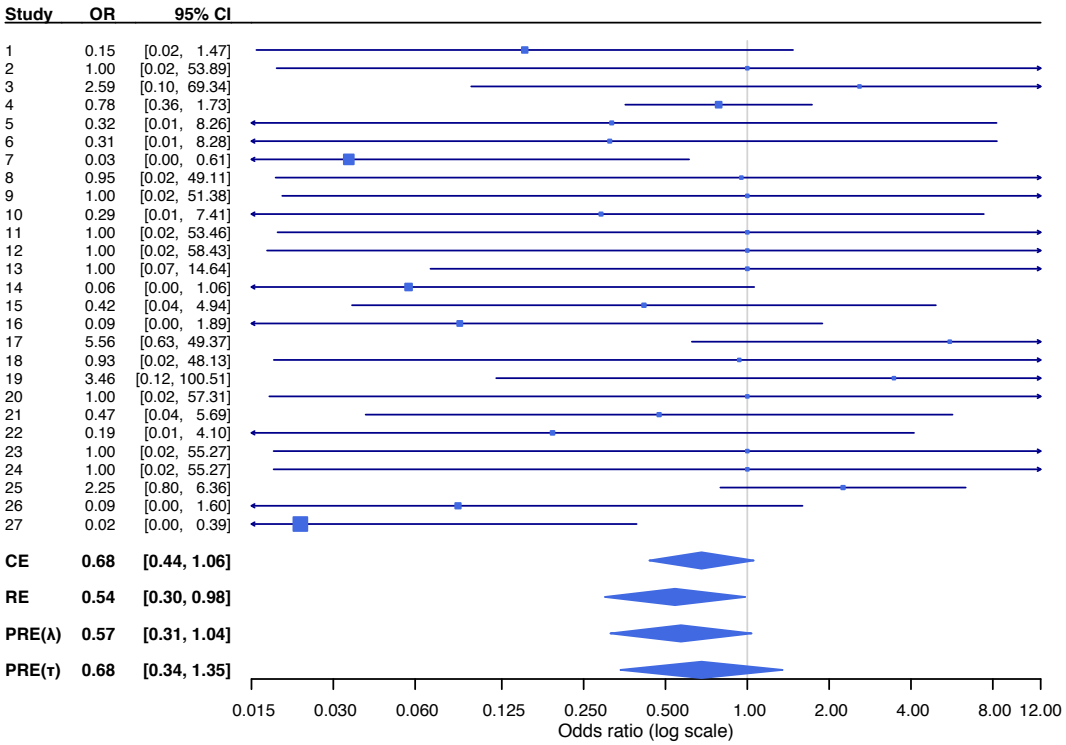


FIGURE S2 Forest plot of the meta-analysis by Keus et al.[6]

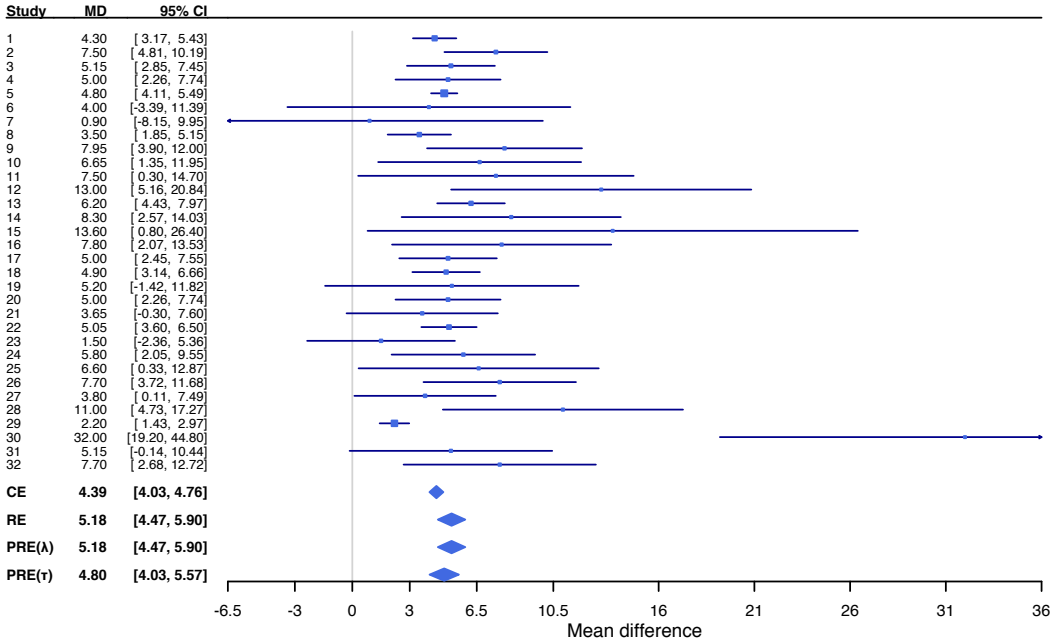


FIGURE S3 Forest plot of the meta-analysis by Adams et al.[7]

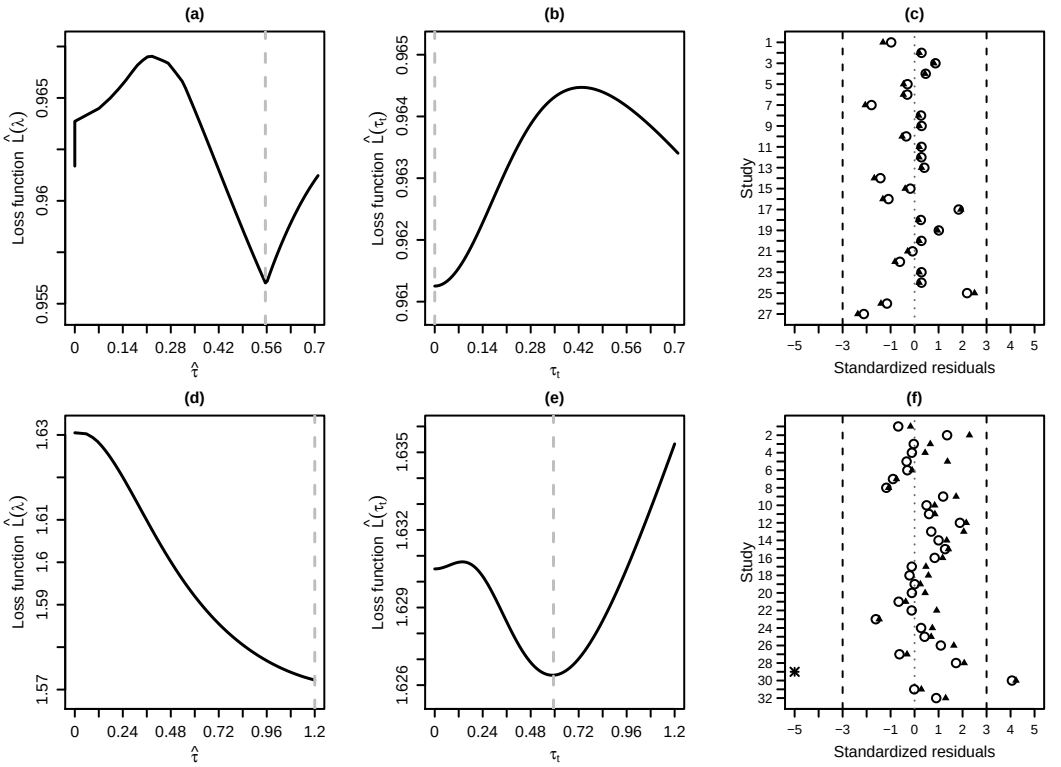


FIGURE S4 Plots for the two additional examples of the meta-analysis by Keus et al.[6] (upper panels) and the meta-analysis by Adams et al.[7] (lower panels). Panels (a) and (d) show the loss function by tuning λ against the estimated between-study standard deviation. Panels (b) and (e) show the loss function by tuning τ against τ_t. Each vertical dashed line represents the optimal value that yields the minimum loss. Panels (c) and (f) present study-specific standardized residuals. Filled triangles represent standardized residuals under the common-effect setting, and unfilled dots represent those under the random-effects setting. Plus and cross signs represent truncated standardized residuals whose absolute values are greater than 5 under the CE and RE settings, respectively.

APPENDIX E. R CODE TO IMPLEMENT THE PENALIZATION METHOD

This appendix contains the R code to produce the results of all empirical data analyses and the simulation study in the main content. Datasets of the examples are also included within the code. The R code consists of three main parts: R functions to construct and implement the penalization method by tuning λ and by tuning τ; analyses of real datasets; and analyses of simulated data.

Specifically, the following code contains five main R functions as follows.

- `metapen.lambda()`, which is used to estimate the between-study variance and the overall effect size for a given λ;
- `metapen.lamb()`, which produces results of the penalization method by tuning λ, and can also provide results of CE and RE models;
- `metapen.lambdas.cv()`, which generates results along with candidate values of λ, and these results can be used to illustrate relationships among different parameters in the penalization method by tuning λ;
- `metapen.tau()`, which produces results of both the CE and RE models, and the penalization method by tuning the between-study standard deviation;
- `forest_reverse_log()`, which generates a forest plot based on the results from `metapen.lamb()` and `metapen.tau()`, and all methods' estimates of the overall effect size and their 95% CI are visualized.

The usage of these functions is provided in the R comments (after the number signs "#" in the following code) and is demonstrated using real and simulated data analyses.

```
## Set your working directory
wd <- "..."

############################
#### Functions
############################

## metaml( ) is used to obtain the ML estimate of the between-study variance.
## arguments: y and s2 represent the observed effect size vector
##    and the corresponding within-study variance vector for a meta-analysis;
##    tol is the relative convergence tolerance in the optimization.
metaml <- function(y, s2, tol = 10^(-10)) {
  if (length(y) != length(s2) | any(s2 < 0))
    stop("error in the input data.")
  if (var(y) == 0)
    stop(return(0))

  mu.hat <- function(tau2) {
    sum(y/(s2 + tau2))/sum(1/(s2 + tau2))
  }
  mu.hat <- Vectorize(mu.hat)

  target.ml <- function(tau2) {
    mean(log(s2 + tau2) + (y - mu.hat(tau2))^2/(s2 + tau2))
  }
  target.ml <- Vectorize(target.ml)

  target1.ml <- function(tau2) {
    p1 <- mean(1/(s2 + tau2))
    p2 <- mean((y - mu.hat(tau2))^2/(s2 + tau2)^2)
    out <- p1 - p2
    return(out)
  }
  target1.ml <- Vectorize(target1.ml)

  tau2.ml <- optim(par = var(y), fn = target.ml, gr = target1.ml, method = "Brent",
                   lower = 0, upper = 100 * var(y), control = list(reltol = tol))$par
  return(tau2.ml)
}

## metareml() is used to obtain the REML estimate of the between-study variance.
```

```
metareml <- function(y, s2, tol = 10^(-10)) {
  if (length(y) != length(s2) | any(s2 < 0))
    stop("error in the input data.")
  if (var(y) == 0)
    stop(return(0))

  mu.hat <- function(tau2) {
    sum(y/(s2 + tau2))/sum(1/(s2 + tau2))
  }

  tau2.old <- 0
  tau2.new <- metaml(y, s2, tol = tol)
  while(abs(tau2.new - tau2.old) > tol){
    tau2.old <- tau2.new
    w <- 1/(s2 + tau2.old)
    temp <- sum(w^2 * ((y - mu.hat(tau2.old))^2 - s2))/sum(w^2) + 1/sum(w)
    tau2.new <- max(0, temp)
  }

  return(tau2.new)
}

## determine.lambda( ) returns candidate values of lambda beteween 0 and lam.c*lambda_max,
##    where lambda_max is the threshold value for lambda.
## arguments: penalty defines the penalty function used in the penalized likelihood, which
##    can be "tau2" (the between-study variance, the penalty we used in the proposed method),
##    "tau" (the between-study standard deviation);
##    n.lambda is the number of candidate values of lambda;
##    lambda.scale represents the transformed scale of lambda, it can be "linear" or "log";
##    lam.c is a scalar used to multiply lambda_max, by controlling its magnitude we can
##    specify the range of candidate values of lambda (the lower bound is 0 and the upper
##    bound is lam.c*lambda_max).
determine.lambda <- function(y, s2, penalty = "tau2", n.lambda = 100,
                             lambda.scale = "log", tol = 10^(-10),
                             lam.c = 1.2){
  if (length(y) != length(s2) | any(s2 < 0))
    stop("error in the input data.")

  mu.hat <- function(tau2){
    sum(y/(s2 + tau2))/sum(1/(s2 + tau2))
  }
  mu.hat <- Vectorize(mu.hat)

  target.ml <- function(tau2){
    mean(log(s2 + tau2) + (y - mu.hat(tau2))^2/(s2 + tau2))
  }
  target.ml <- Vectorize(target.ml)

  target1.ml <- function(tau2){
    p1 <- mean(1/(s2 + tau2))
    p2 <- mean((y - mu.hat(tau2))^2/(s2 + tau2)^2)
    out <- p1 - p2
    return(out)
  }
  target1.ml <- Vectorize(target1.ml)

  tau2.ml <- optim(par = var(y), fn = target.ml, gr = target1.ml, method = "Brent",
                   lower = 0, upper = 100*var(y), control = list(reltol = tol))$par

  if(penalty == "tau2"){
    fcn <- function(tau2){
      sum((y - mu.hat(tau2))^2/(s2 + tau2)^2) - sum(1/(s2 + tau2))
    }
  }
  if(penalty == "tau"){
    fcn <- function(tau2){
      (sum((y - mu.hat(tau2))^2/(s2 + tau2)^2) - sum(1/(s2 + tau2)))*(2*sqrt(tau2))
    }
  }
  fcn <- Vectorize(fcn)
  lam.max <- optimize(f = fcn, lower = 0, upper = tau2.ml*1.1, maximum = TRUE)$objective
  if(lambda.scale == "linear") lam.max <- max(c(0, lam.max))*lam.c
  if(lambda.scale == "log") lam.max <- exp(log(max(c(0, lam.max)) + 1) + log(lam.c)) - lam.c
```

```
  if(lambda.scale == "linear") lambda <- seq(from = 0, to = lam.max, length.out = n.lambda)
  if(lambda.scale == "log"){
    lambda <- exp(seq(from = 0, to = log(lam.max + 1), length.out = n.lambda)) - 1
  }

  return(lambda)
}

## metapen.lambda( ) compute the overall effect size estimates and the between-study
## variance estimates for a vector of lambda.
## arguments: lambda is a vector of candidate values (e.g., the result of determine.lambda());
##    tau2.ml is the ML estimate of the between-study variance, it can be computed by the function
##    itself if not provided;
metapen.lambda <- function(y, s2, penalty = "tau2", lambda, tau2.ml, tol = 10^(-10)){
  if (length(y) != length(s2) | any(s2 < 0))
    stop("error in input data.")
  I <- length(y)

  if(!is.element(penalty, c("tau", "tau2")))
    stop("penalty must be specified as 'tau' or 'tau2'.")

  if(penalty == "tau2"){
    pen <- function(tau2) {tau2}
    pen1 <- function(tau2) {1}
  }else{
    pen <- function(tau2) {sqrt(tau2)}
    pen1 <- function(tau2) {1/sqrt(2 * tau2)}
  }
  pen <- Vectorize(pen)
  pen1 <- Vectorize(pen1)

  mu.hat <- function(tau2){
    sum(y/(s2 + tau2))/sum(1/(s2 + tau2))
  }
  mu.hat <- Vectorize(mu.hat)

  if(missing(tau2.ml)){
    target.ml <- function(tau2){
      mean(log(s2 + tau2) + (y - mu.hat(tau2))^2/(s2 + tau2))
    }
    target.ml <- Vectorize(target.ml)

    target1.ml <- function(tau2){
      p1 <- mean(1/(s2 + tau2))
      p2 <- mean((y - mu.hat(tau2))^2/(s2 + tau2)^2)
      out <- p1 - p2
      return(out)
    }
    target1.ml <- Vectorize(target1.ml)

    tau2.ml <- optim(par = var(y), fn = target.ml, gr = target1.ml, method = "Brent",
                     lower = 0, upper = 100*var(y), control = list(reltol = tol))$par
  }

  tau2.hat <- function(lambda){
    target <- function(tau2){
      mean(log(s2 + tau2) + (y - mu.hat(tau2))^2/(s2 + tau2)) + lambda*pen(tau2)/I
    }
    target <- Vectorize(target)

    target1 <- function(tau2){
      p1 <- mean(1/(s2 + tau2))
      p2 <- mean((y - mu.hat(tau2))^2/(s2 + tau2)^2)
      out <- p1 - p2 + lambda*pen1(tau2)/I
      return(out)
    }

    out <- optim(par = tau2.ml, fn = target, gr = target1, method = "Brent",
                 lower = 0, upper = tau2.ml*1.1, control = list(reltol = tol))$par
    return(out)
  }
  tau2.hat <- Vectorize(tau2.hat)

  out.tau2 <- tau2.hat(lambda)
```

```
  out.mu <- mu.hat(out.tau2)
  out <- cbind(lambda = lambda, mu = out.mu, tau2 = out.tau2)
  rownames(out) <- rep("", length(lambda))
  return(out)
}

## metapen.lambdas.cv() generates results along with candidate values of lambda
## arguments: re.method represents the method to estimate the between-study variance
##   of the random-effects model, which can be "ML" or "REML".
metapen.lambdas.cv <- function(y, s2, penalty = "tau2", re.method = "ML",
                               lambda, n.lambda, lambda.scale = "log",
                               tau2.ml, tol = 10^(-10), lam.c = 1.2){
  if(length(y) != length(s2) | any(s2 < 0))
    stop("error in input data.")
  I <- length(y)

  if(!is.element(lambda.scale, c("log", "linear")))
    stop("lambda.scale must be either 'linear' or 'log'.")

  if(!missing(lambda) & !missing(n.lambda)){
    if(n.lambda != length(lambda))
      stop("mismatched lambda and n.lambda.")
  }

  if(!missing(lambda)){
    n.lambda <- length(lambda)
  }

  ## If lam.c = 1, the range of lambda is between 0 and lambda_max
  if(missing(lambda)){
    if(missing(n.lambda)) n.lambda <- 100
    lambda <- determine.lambda(y = y, s2 = s2, penalty = penalty,
                               n.lambda = n.lambda, lambda.scale = lambda.scale, tol = tol,
                               lam.c = lam.c)
  }

  diff <- matrix(NA, n.lambda, I)
  for(i in 1:I){
    y_i <- y[-i]
    s2_i <- s2[-i]
    w_i <- 1/s2_i
    if(re.method == "ML"){
      tau2.re_i <- metaml(y_i, s2_i, tol)
    }
    if(re.method == "REML"){
      tau2.re_i <- metareml(y_i, s2_i, tol)
    }
    out <- metapen.lambda(y = y_i, s2 = s2_i, lambda = lambda,
                          penalty = penalty, tol = tol)
    mu_i <- out[, "mu"]
    tau2_i <- out[, "tau2"]
    mu.hat.var <- lapply(X = 1:n.lambda, FUN = function(lam) {
      sum((s2_i + tau2.re_i)/(s2_i + tau2_i[lam])^2)/(sum(1/(s2_i + tau2_i[lam])))^2
    })
    mu.hat.var <- unlist(mu.hat.var)
    diffi <- (y[i] - mu_i)^2/(s2[i] + tau2.re_i + mu.hat.var)
    diff[, i] <- diffi
  }

  ## With the full dataset, using metapen.lambda function again to find tau^2 and mu
  ## for a given lambda
  if(missing(tau2.ml)){
    full <- metapen.lambda(y = y, s2 = s2, lambda = lambda, penalty = penalty,
                           tol = tol)
  }else{
    full <- metapen.lambda(y = y, s2 = s2, lambda = lambda, penalty = penalty,
                           tau2.ml = tau2.ml, tol = tol)
  }
  mu <- full[, "mu"]
  tau2 <- full[, "tau2"]
  tau<- sqrt(tau2)

  loss <- apply(diff, 1, mean)
  loss <- sqrt(loss)
```

```
  est <- cbind(lambda, loss, mu, tau2, tau)
  rownames(est) <- rep("", n.lambda)

  out <- list(est = est, penalty = penalty, n.lambda = n.lambda,
              lambda.scale = lambda.scale, tol = tol)
  return(out)
}

## metapen() conducting the penalization method by tuning lambda or by
## tuning tau (the between-study standard deviation).
## arguments: tuning.way represents the tuning parameter is used in the
##   penalization method, it has two levels, "tau" and "lambda".
metapen <- function(y, s2, re.method = "ML", tuning.way = "tau", upp,
                    n.cand = 100, penalty = "tau2", lambda, n.lambda,
                    lambda.scale = "log", tau2.ml, lam.c = 1.2,
                    tol = 10^(-10)){
  if(length(y) != length(s2) | any(s2 < 0))
    stop("error in input data.")
  I <- length(y)

  if(tuning.way == "tau"){
    w <- 1/s2
    y.bar <- sum(w*y)/sum(w)
    Q <- sum(w*(y - y.bar)^2)
    if(Q < I - 1) Q <- I - 1
    I2 <- (Q - (I - 1))/Q

    if(re.method == "ML"){
      tau2.re <- metaml(y, s2, tol)
    }
    if(re.method == "REML"){
      tau2.re <- metareml(y, s2, tol)
    }

    mu.fe <- y.bar
    se.fe <- sqrt(1/sum(w))
    w.re <- 1/(s2 + tau2.re)
    mu.re <- sum(w.re * y)/sum(w.re)
    se.re <- sqrt(1/sum(w.re))

    if(missing(upp)) upp <- 1
    tau2.upp <- upp * tau2.re

    tau.cand <- seq(from = 0, to = sqrt(tau2.upp), length.out = n.cand)
    tau2.cand <- tau.cand^2

    errs <- matrix(0, I, n.cand)
    for(i in 1:I){
      y.train <- y[-i]
      s2.train <- s2[-i]
      if(re.method == "ML"){
        tau2.re.train <- metaml(y.train, s2.train, tol)
      }
      if(re.method == "REML"){
        tau2.re.train <- metareml(y.train, s2.train, tol)
      }
      for(j in 1:n.cand){
        tau2.temp <- tau2.cand[j]
        w.temp <- 1/(s2.train + tau2.temp)
        mu_i <- sum(w.temp * y.train)/sum(w.temp)
        var_i <- sum(w.temp^2 * (s2.train + tau2.re.train))/(sum(w.temp))^2
        errs[i, j] <- (mu_i - y[i])^2/(var_i + s2[i] + tau2.re.train)
      }
    }
    loss <- sqrt(colMeans(errs))
    opt.idx <- which(loss == min(loss))
    opt.idx <- opt.idx[1]  ## If several cases have the same minimize loss, select the first case
    tau2.opt <- tau2.cand[opt.idx]
    tau.opt <- tau.cand[opt.idx]
    w.opt <- 1/(s2 + tau2.opt)
    mu.opt <- sum(w.opt * y)/sum(w.opt)
    var.opt <- sum(w.opt^2 * (s2 + tau2.re))/(sum(w.opt))^2

    out <- NULL
```

```
    out$n.study <- I
    out$tau2.re <- tau2.re
    out$I2 <- I2
    out$mu.fe <- mu.fe
    out$se.fe <- se.fe
    out$mu.re <- mu.re
    out$se.re <- se.re
    out$tau2.cand <- tau2.cand
    out$tau.cand <- tau.cand
    out$loss <- loss
    out$tau2.opt <- tau2.opt
    out$tau.opt <- tau.opt  ## The estimate of the between-study standard deviation
    out$mu.opt <- mu.opt  ## The estimate of the overall effect size
    out$se.opt <- sqrt(var.opt)  ## The standard error of the overall effect size estimate
  }

  if(tuning.way == "lambda"){
    if(length(y) != length(s2) | any(s2 < 0))
      stop("error in input data.")
    I <- length(y)

    if(!is.element(lambda.scale, c("log", "linear")))
      stop("lambda.scale must be either 'linear' or 'log'.")

    if(!missing(lambda) & !missing(n.lambda)){
      if(n.lambda != length(lambda))
        stop("mismatched lambda and n.lambda.")
    }

    if(!missing(lambda)){
      n.lambda <- length(lambda)
    }

    ## If lam.c = 1, the range of lambda is between 0 and lambda_max
    if(missing(lambda)){
      if(missing(n.lambda)) n.lambda <- 100
      lambda <- determine.lambda(y = y, s2 = s2, penalty = penalty,
                n.lambda = n.lambda, lambda.scale = lambda.scale, tol = tol, lam.c = lam.c)
    }

    w <- 1/s2
    y.bar <- sum(w*y)/sum(w)
    Q <- sum(w*(y - y.bar)^2)
    I2 <- (Q - (I - 1))/Q

    if(re.method == "ML"){
      tau2.re <- metaml(y, s2, tol)
    }
    if(re.method == "REML"){
      tau2.re <- metareml(y, s2, tol)
    }

    mu.fe <- y.bar
    se.fe <- sqrt(1/sum(w))
    w.re <- 1/(s2 + tau2.re)
    mu.re <- sum(w.re*y)/sum(w.re)
    se.re <- sqrt(1/sum(w.re))

    diff <- matrix(NA, n.lambda, I)
    for(i in 1:I){
      y_i <- y[-i]
      s2_i <- s2[-i]
      w_i <- 1/s2_i
      if(re.method == "ML"){
        tau2.re_i <- metaml(y_i, s2_i, tol)
      }
      if(re.method == "REML"){
        tau2.re_i <- metareml(y_i, s2_i, tol)
      }
      out <- metapen.lambda(y = y_i, s2 = s2_i, lambda = lambda, penalty = penalty,
                            tol = tol)
      mu_i <- out[,"mu"]
      tau2_i <- out[,"tau2"]
      mu.hat.var <- lapply(X = 1:n.lambda, FUN = function(lam) {
```

```
        sum((s2_i + tau2.re_i)/(s2_i + tau2_i[lam])^2)/(sum(1/(s2_i + tau2_i[lam])))^2
      })
      mu.hat.var <- unlist(mu.hat.var)
      diffi <- (y[i] - mu_i)^2/(s2[i] + tau2.re_i + mu.hat.var)
      diff[,i] <- diffi
    }

    ## With the full dataset, using metapen.lambda( ) to estimate tau^2 and mu for a given lambda
    if(missing(tau2.ml)){
      full <- metapen.lambda(y = y, s2 = s2, lambda = lambda, penalty = penalty,
                             tol = tol)
    }else{
      full <- metapen.lambda(y = y, s2 = s2, lambda = lambda, penalty = penalty,
                             tau2.ml = tau2.ml, tol = tol)
    }
    mu <- full[, "mu"]
    tau2 <- full[, "tau2"]
    tau<- sqrt(tau2)

    loss <- apply(diff, 1, mean)
    loss <- sqrt(loss)
    opt.idx <- which(loss == min(loss))
    opt.idx <- opt.idx[1]   ## If several cases have the same minimize loss, select the first case
    tau2.opt <- tau2[opt.idx]
    tau.opt <- tau[opt.idx]
    w.opt <- 1/(s2 + tau2.opt)
    mu.opt <- sum(w.opt * y)/sum(w.opt)
    var.opt <- sum(w.opt^2 * (s2 + tau2.re))/(sum(w.opt))^2

    out <- NULL
    out$n.study <- I
    out$tau2.re <- tau2.re
    out$I2 <- I2
    out$mu.fe <- mu.fe
    out$se.fe <- se.fe
    out$mu.re <- mu.re
    out$se.re <- se.re
    out$tau2 <- tau2
    out$tau <- tau
    out$loss <- loss
    out$tau2.opt <- tau2.opt
    out$tau.opt <- tau.opt  ## The estimate of the between-study standard deviation
    out$mu.opt <- mu.opt  ## The estimate of the overall effect size
    out$se.opt <- sqrt(var.opt)  ## The standard error of the overall effect size estimate
  }

  return(out)
}

## metaoutliers.plot( ) plots study-specific standardized residuals for a given meta-analysis.
metaoutliers.plot<- function(y, s2, re.method = "ML"){
  if(length(y) != length(s2))
    stop("error in input data.")
  I <- length(y)

  ## Compute the standardized residuals under the common-effect setting
  w <- 1/s2
  mu.hat.i <- e <- e.tilde.fe <- numeric(I)
  for(i in 1:I){
    w.temp <- w[-i]
    y.temp <- y[-i]
    mu.hat.i[i] <- sum(y.temp*w.temp)/sum(w.temp)
    e[i] <- y[i] - mu.hat.i[i]
    sig2.e.i <- 1/sum(w.temp) + 1/w[i]
    e.tilde.fe[i] <- e[i]/sqrt(sig2.e.i)
  }

  ## Compute the standardized residuals under the random-effects setting
  mu.hat.i <- e <- e.tilde.re <- numeric(I)
  for(i in 1:I){
    s2.temp <- s2[-i]
    y.temp <- y[-i]
    if(re.method == "ML"){
      tau2.temp <- metaml(y.temp, s2.temp)
```

```
    }
    if(re.method == "REML"){
      tau2.temp <- metareml(y.temp, s2.temp)
    }
    w.temp <- 1/(s2.temp + tau2.temp)
    mu.hat.i[i] <- sum(y.temp*w.temp)/sum(w.temp)
    e[i] <- y[i] - mu.hat.i[i]
    var.e.i <- 1/sum(w.temp) + s2[i] + tau2.temp
    e.tilde.re[i] <- e[i]/sqrt(var.e.i)
  }

  std.res <- c(e.tilde.fe, e.tilde.re)

  shapes <- NULL
  for(i in 1:I){
    if (std.res[i] <= -5){shapes[i] = 3; std.res[i] = -5}
    else if (std.res[i] > -5 & std.res[i] < 5){shapes[i] = 17}
    else {shapes[i] = 3; std.res[i] = 5}
  }

  for(i in (I+1):(2*I)){
    if (std.res[i] <= -5){shapes[i] = 4; std.res[i] = -5}
    else if ( std.res[i] > -5 & std.res[i] < 5){shapes[i] = 1}
    else {shapes[i] = 4; std.res[i] = 5}
  }

  par(mar = c(2.4, 2.8, 1, 1) + 0.1)
  plot(std.res[1:I], I:1, ylim = c(1, I), xlim = c(-5, 5),
       yaxt = "n", xaxt = "n", pch = shapes[1:I], cex = 0.6)
  title(xlab = "Standardized residuals", ylab = "Study", cex.lab = 0.8, line = 1.1)
  axis(side = 2, at = 1:I, labels = I:1, tck= -0.03, cex.axis = 0.6, las = 1, mgp = c(3, 0.5, 0))
  axis(side = 1, at = -5:5, tck = -0.03, cex.axis = 0.6, mgp = c(3, 0.2, 0))
  abline(v = c(-3, 3), lwd = 1, lty = 2)
  abline(v = 0, lwd = 1, lty = 3, col = "dimgrey")

  points(std.res[(I + 1):(2 * I)], I:1, pch = shapes[(I + 1):(2 * I)], cex = 0.9)

}

## forest_reverse_log( ) produce a forestplot for a given meta-analysis,
## it summarises all methods' estimates and their 95% confidence intervals,
## this function requires two R packages: "meta" and "forestplot".
## arguments: y is the observed effect size that measured on the log scale
##    (e.g., log odds ratio and log relative risk);
##    s2 is the corresponding within-study variance;
##    out is the summary outcome of the metagen( ) in the R package meta;
##    adjust_font_size is a scalar used to control the font size of labels and ticks;
##    llimit and ulimit represent lower and upper limits for clipping
##    confindence intevals to arrows;
##    user-specified x-axis tick marks are used if user_define_x_ticks is TRUE,
##    otherwise (i.e., user_define_x_ticks is FALSE) defaults are used;
##    x_ticks represent user-specified x-axis tick marks, it must be provided when
##    user_define_x_ticks is TRUE;
##    x_ticks are not specified when user_define_x_ticks is FALSE, and the plot
##    will use the default values;
##    effect_type represents the type of effect sizes that are reported on the log scale,
##    which can be "OR" (odds ratio) or "RR" (relative risk);
##    way represent the method using to estimate between-study variance.
forest_reverse_log <- function(y, s2, out, adjust_font_size = 0, llimit = 0.25,
                               ulimit = 16, user_define_x_ticks = FALSE, x_ticks,
                               effect_type = "OR", re.method = "ML"){
  if(length(y) != length(s2) | any(s2 < 0))
    stop("Errors in the input data.")
  if(abs(adjust_font_size) > 1)
    stop("The adjusted font size cannot be unreasonable")
  if(missing(out)){
    out <- metagen(y, sqrt(s2), sm = effect_type, method.tau = re.method)
  }

  I <- length(y)

  eff.size <- exp(y)
  eff.upper <- exp(out$upper)
  eff.lower <- exp(out$lower)
```

```
tradeoff.lambda <- metapen(y, s2, re.method, tuning.way = "lambda",
                           n.lambda = 100, lam.c = 1.2)
lower.lambda <- exp(tradeoff.lambda$mu.opt - 1.96*(tradeoff.lambda$se.opt))
upper.lambda <- exp(tradeoff.lambda$mu.opt + 1.96*(tradeoff.lambda$se.opt))

tradeoff.tau <- metapen(y, s2, re.method, tuning.way = "tau")
lower.tau <- exp(tradeoff.tau$mu.opt - 1.96*(tradeoff.tau$se.opt))
upper.tau <- exp(tradeoff.tau$mu.opt + 1.96*(tradeoff.tau$se.opt))

gap <- as.integer(-(I+10))

units <- structure(list(mean = c(NA, NA, eff.size, NA, exp(out$TE.fixed), NA,
                  exp(out$TE.random), NA, exp(tradeoff.lambda$mu.opt),
                  NA, exp(tradeoff.tau$mu.opt)), lower = c(NA, NA, eff.lower, NA,
                  exp(out$lower.fixed), NA, exp(out$lower.random), NA, lower.lambda,
                  NA, lower.tau), upper = c(NA, NA, eff.upper, NA, exp(out$upper.fixed),
                  NA, exp(out$upper.random), NA, upper.lambda, NA, upper.tau)),
                  .Names = c("mean", "lower", "upper"), row.names = c(NA, gap),
                  class = "data.frame")

eff.round <- format(round(eff.size, digit=2), nsmall=2)
fix.round <- format(round(exp(out$TE.fixed), digit=2), nsmall=2)
ran.round <- format(round(exp(out$TE.random), digit=2), nsmall=2)
trdlam.round <- format(round(exp(tradeoff.lambda$mu.opt), digit=2), nsmall=2)
trdtau.round <- format(round(exp(tradeoff.tau$mu.opt), digit=2), nsmall=2)
upper.round <- format(round(eff.upper, digit=2), nsmall=2)
lower.round <- format(round(eff.lower, digit=2), nsmall=2)
fix.upper.round <- format(round(exp(out$upper.fixed), digit=2), nsmall=2)
fix.lower.round <- format(round(exp(out$lower.fixed), digit=2), nsmall=2)
ran.upper.round <- format(round(exp(out$upper.random), digit=2), nsmall=2)
ran.lower.round <- format(round(exp(out$lower.random), digit=2), nsmall=2)
trdlam.upper.round <- format(round(upper.lambda, digit=2), nsmall=2)
trdlam.lower.round <- format(round(lower.lambda, digit=2), nsmall=2)
trdtau.upper.round <- format(round(upper.tau, digit=2), nsmall=2)
trdtau.lower.round <- format(round(lower.tau, digit=2), nsmall=2)

CI <- paste("[", lower.round, ",", " ", upper.round, "]", sep="")
CI.fixed <- paste("[", fix.lower.round, ",", " ", fix.upper.round, "]", sep="")
CI.random <- paste("[", ran.lower.round, ",", " ", ran.upper.round, "]", sep="")
CI.tradeoff.lambda <- paste("[", trdlam.lower.round, ",", " ", trdlam.upper.round, "]", sep="")
CI.tradeoff.tau <- paste("[", trdtau.lower.round, ",", " ", trdtau.upper.round, "]", sep="")

tabletext <- cbind(c("Study", NA, as.character(out$studlab), NA, "CE",
                     NA, "RE", NA, paste("PRE", "(", intToUtf8(955), ")", sep = ""),
                     NA, paste("PRE", "(", intToUtf8(964), ")", sep = "")),
                   c(effect_type, NA, eff.round, NA, fix.round, NA, ran.round,
                     NA, trdlam.round, NA, trdtau.round),
                   c("95% CI", NA, CI, NA, CI.fixed, NA, CI.random,
                     NA, CI.tradeoff.lambda, NA, CI.tradeoff.tau))

space <- I+2

if(effect_type == "OR"){x_label <- "Odds ratio (log scale)"}
if(effect_type == "RR"){x_label <- "Risk ratio (log scale)"}

if(user_define_x_ticks == FALSE){
  forestplot(tabletext, hrzl_lines = list("2" = gpar(lty=1)), units, new_page = F,
             is.summary=c(TRUE, rep(FALSE, space), TRUE, FALSE, TRUE, FALSE,
                          TRUE, FALSE, TRUE),
             txt_gp = fpTxtGp( label = gpar(fontfamily = "", cex = .6 + adjust_font_size),
                               ticks = gpar(fontfamily = "", cex = .65 + adjust_font_size),
                               xlab = gpar(fontfamily = "", cex = .7 + adjust_font_size)),
             xlog = T, clip = c(llimit, ulimit), colgap = unit(3, "mm"),
             line.margin = .35,
             col = fpColors(box = "royalblue", line = "darkblue", summary ="royalblue"),
             xlab = x_label, mar = unit(c(.01, 3, 2, 3.5), "mm"))
} else {
  forestplot(tabletext, hrzl_lines = list("2" = gpar(lty=1)), units, new_page = F,
             is.summary=c(TRUE, rep(FALSE, space), TRUE, FALSE, TRUE, FALSE,
                          TRUE, FALSE, TRUE),
             txt_gp = fpTxtGp( label = gpar(fontfamily = "", cex = .6 + adjust_font_size),
                               ticks = gpar(fontfamily = "", cex = .65 + adjust_font_size),
                               xlab = gpar(fontfamily = "", cex = .7 + adjust_font_size)),
```

```
                xlog = T, xticks = x_ticks, clip = c(llimit, ulimit),
                colgap = unit(3, "mm"), line.margin = .35,
                col = fpColors(box = "royalblue", line = "darkblue", summary ="royalblue"),
                xlab = x_label, mar = unit(c(.01, 3, 2, 3.5), "mm"))
  }

}

## forest_original_effects() produces a forest plot for the meta-analysis
## whose effects are reported directly (not on the log scale) with the
## corresponding variance estimates.
## arguments: effect_type represents the type of effect sizes, which can be
##    "MD" (mean difference), "SMD" (standardized mean difference), "RD"
##     (risk difference).
forest_original_effects <- function(y, s2, out, adjust_font_size = 0, llimit = -10,
                                    ulimit = 20, user_define_x_ticks = FALSE, x_ticks,
                                    effect_type = "MD", re.method = "ML"){
  if(length(y) != length(s2) | any(s2 < 0))
    stop("Errors in the input data.")
  if(abs(adjust_font_size) > 1)
    stop("The adjusted font size cannot be unreasonable")
  if(missing(out)){
    out <- metagen(y, sqrt(s2), sm = effect_type, method.tau = re.method)
  }

  I <- length(y)

  eff.size <- y
  eff.upper <- out$upper
  eff.lower <- out$lower

  tradeoff.lambda <- metapen(y, s2, re.method, tuning.way = "lambda",
                             n.lambda = 100, lam.c = 1.2)
  lower.lambda <- tradeoff.lambda$mu.opt - 1.96*(tradeoff.lambda$se.opt)
  upper.lambda <- tradeoff.lambda$mu.opt + 1.96*(tradeoff.lambda$se.opt)

  tradeoff.tau <- metapen(y, s2, re.method, tuning.way = "tau")
  lower.tau <- tradeoff.tau$mu.opt - 1.96*(tradeoff.tau$se.opt)
  upper.tau <- tradeoff.tau$mu.opt + 1.96*(tradeoff.tau$se.opt)

  gap <- as.integer(-(I+10))

  units <- structure(list(mean = c(NA, NA, eff.size, NA, out$TE.fixed, NA,
                       out$TE.random, NA, tradeoff.lambda$mu.opt, NA, tradeoff.tau$mu.opt),
                       lower = c(NA, NA, eff.lower, NA, out$lower.fixed, NA, out$lower.random,
                       NA, lower.lambda, NA, lower.tau), upper = c(NA, NA, eff.upper, NA,
                       out$upper.fixed, NA, out$upper.random, NA, upper.lambda, NA, upper.tau)),
                       .Names = c("mean", "lower", "upper"), row.names = c(NA, gap),
                       class = "data.frame")

  eff.round <- format(round(eff.size, digit=2), nsmall=2)
  fix.round <- format(round(out$TE.fixed, digit=2), nsmall=2)
  ran.round <- format(round(out$TE.random, digit=2), nsmall=2)
  trdlam.round <- format(round(tradeoff.lambda$mu.opt, digit=2), nsmall=2)
  trdtau.round <- format(round(tradeoff.tau$mu.opt, digit=2), nsmall=2)
  upper.round <- format(round(eff.upper, digit=2), nsmall=2)
  lower.round <- format(round(eff.lower, digit=2), nsmall=2)
  fix.upper.round <- format(round(out$upper.fixed, digit=2), nsmall=2)
  fix.lower.round <- format(round(out$lower.fixed, digit=2), nsmall=2)
  ran.upper.round <- format(round(out$upper.random, digit=2), nsmall=2)
  ran.lower.round <- format(round(out$lower.random, digit=2), nsmall=2)
  trdlam.upper.round <- format(round(upper.lambda, digit=2), nsmall=2)
  trdlam.lower.round <- format(round(lower.lambda, digit=2), nsmall=2)
  trdtau.upper.round <- format(round(upper.tau, digit=2), nsmall=2)
  trdtau.lower.round <- format(round(lower.tau, digit=2), nsmall=2)

  CI <- paste("[", lower.round, ",", " ", upper.round, "]", sep="")
  CI.fixed <- paste("[", fix.lower.round, ",", " ", fix.upper.round, "]", sep="")
  CI.random <- paste("[", ran.lower.round, ",", " ", ran.upper.round, "]", sep="")
  CI.tradeoff.lambda <- paste("[", trdlam.lower.round, ",", " ", trdlam.upper.round, "]", sep="")
  CI.tradeoff.tau <- paste("[", trdtau.lower.round, ",", " ", trdtau.upper.round, "]", sep="")

  tabletext <- cbind(c("Study", NA, as.character(out$studlab), NA, "CE", NA, "RE",
                       NA, paste("PRE", "(", intToUtf8(955), ")", sep=""), NA,
```

```
                      paste("PRE", "(", intToUtf8(964), ")", sep="")), c(effect_type,
                      NA, eff.round, NA, fix.round, NA, ran.round, NA, trdlam.round,
                      NA, trdtau.round), c("95% CI", NA, CI, NA, CI.fixed, NA, CI.random,
                      NA, CI.tradeoff.lambda, NA, CI.tradeoff.tau))

  space <- I+2

  if(effect_type == "MD"){x_label <- "Mean difference"}
  if(effect_type == "SMD"){x_label <- "Standardized mean difference"}
  if(effect_type == "RD"){x_label <- "Risk difference"}

  if(user_define_x_ticks == FALSE){
    forestplot(tabletext, hrzl_lines = list("2" = gpar(lty=1)),units, new_page = F,
               is.summary=c(TRUE, rep(FALSE, space), TRUE, FALSE, TRUE, FALSE,
               TRUE, FALSE, TRUE), txt_gp = fpTxtGp( label = gpar(fontfamily = "",
               cex = .5 + adjust_font_size),ticks = gpar(fontfamily = "",
               cex = .6 + adjust_font_size), xlab = gpar(fontfamily = "",
               cex = .7 + adjust_font_size)), xlog = F, colgap = unit(3, "mm"),
               line.margin = .35, col = fpColors(box = "royalblue", line = "darkblue",
               summary ="royalblue"), xlab = x_label, mar = unit(c(.01, 3, 2, 3.5), "mm")
    )
  }

  if(user_define_x_ticks == TRUE){
    forestplot(tabletext, hrzl_lines = list("2" = gpar(lty=1)), units, new_page = F,
               is.summary=c(TRUE, rep(FALSE, space), TRUE, FALSE, TRUE, FALSE, TRUE, FALSE, TRUE),
               txt_gp = fpTxtGp( label = gpar(fontfamily = "", cex = .5 + adjust_font_size),
                                 ticks = gpar(fontfamily = "", cex = .6 + adjust_font_size),
                                 xlab = gpar(fontfamily = "", cex = .7 + adjust_font_size)),
               xlog = F, xticks = x_ticks, clip = c(llimit, ulimit),
               colgap = unit(3, "mm"), line.margin = .35,
               col = fpColors(box = "royalblue", line = "darkblue", summary ="royalblue"),
               xlab = x_label, mar = unit(c(.01, 3, 2, 3.5), "mm")
    )
  }

}

## The following two functions are based on the paper by Wang and Lee (2020),
## meta() is used to calculate estimates of overall effect size and its variance
## with estimated tau2; sswtest() returns the standardized effect sizes and the
## p-value of the Shapiro-Wilk test.
meta <- function(y, s2, re.method = "ML"){
  num <- length(y)
  ## Notice that sm argument of metagen() is useless
  ## if the effect metric is pre-sepecified
  tau <- metagen(y, sqrt(s2), method.tau = re.method)$tau
  tau2 <- tau^2
  thetar <- sum(y/(s2+tau2))/sum((s2+tau2)^(-1))
  seTE2 <- sum((y-thetar)^2*((s2+tau2)^(-1)/sum((s2+tau2)^(-1))))/(num-1)
  return(list(thetar=thetar, seTE2=seTE2, tau2=tau2))
}

sswtest <- function(y, s2, re.method = "ML"){
  num <- length(y)
  tau2 <- meta(y, s2, re.method)$tau2
  if(any(s2<0)) stop("Standard error(s) < 0")
  if(tau2<=0) stop("The estimated tau2 by the DL method is 0. The common effect model is favored.")
  stdmean <- rep(0, num)
  for (i in 1:num) {
    thetarJ <- meta(y[-i], s2[-i])$thetar
    seTE2J <- meta(y[-i], s2[-i])$seTE2
    stdmean[i] <- (y[i]-thetarJ)*(1/(tau2+seTE2J+s2[i]))^0.5
  }
  pvalue <- shapiro.test(stdmean)$p.value
  result <- list(stdmean=stdmean, pvalue=pvalue)
  return(result)
}



############################
#### Real data analysis
############################
```

```
setwd(wd)

## Load necessary packages
library(meta)
library(forestplot)
library(Cairo)  ## XQuartz is needed to refer Cairo package on mac

## Meta-analysis in Bohren, Meghan A., et al. (2017), CDSR
Bohren <- data.frame(
  n1 = c(50, 105, 291, 3454, 499, 53, 217, 357, 209, 133, 654,
         282, 92, 212, 41, 81, 75, 50, 72, 58, 168),
  n2 = c(50, 107, 295, 3461, 493, 56, 218, 357, 204, 129, 665,
         263, 97, 200, 38, 80, 75, 50, 73, 56, 249),
  r1 = c(46, 41, 223, 2481, 280, 48, 110, 260, 132, 97, 451, 202,
         74, 179, 38, 76, 73, 47, 47, 33, 154),
  r2 = c(38, 38, 220, 2463, 239, 40, 101, 247, 127, 85, 425, 183,
         76, 137, 34, 72, 74, 30, 42, 31, 196))
## Compute the log odds ratios and the corresponding variances
n1 <- Bohren$n1; n2 <- Bohren$n2; r1 <- Bohren$r1; r2 <- Bohren$r2; c1 <- n1 - r1; c2 <- n2 - r2
## Use Haldane-Anscombe correction to adjust the cell counts if there exist
## any cell equals to zero
L <- length(n1)
for(j in 1:L){
  if(r1[j]==0 | r2[j]==0 | c1[j]==0 | c2[j]==0){
    r1[j] <- r1[j] + 0.5; r2[j] <- r2[j] + 0.5; c1[j] <- c1[j] + 0.5; c2[j] <- c2[j] + 0.5}
}
odr <- (r1*c2) / (r2*c1)
Bohren$y <- log(odr)
Bohren$s2 <- 1/r1 + 1/c1 + 1/r2 +1/c2
print(Bohren)
y <- Bohren$y; s2 <- Bohren$s2

out.Bohren <- metagen(y, sqrt(s2), sm = "OR", method.tau = "ML")
print(out.Bohren)

## Figure 1(a) in the main content: Forest plot of the meta-analysis by Bohren et al. (2017).
cairo_pdf(filename = "Figure1a.pdf", width= 6.33, height= 3.67)
forest_reverse_log(y, s2, llimit = 0.25, ulimit = 16, user_define_x_ticks = TRUE,
                   x_ticks = c(0.25, 0.5, 1, 2, 4, 8, 16), out = out.Bohren, re.method = "ML")
dev.off()

## Compute the standardized residuals of the common-effect model
w <- 1/s2
mu.hat.i <- e <- e.tilde.fe <- numeric(L)
for(i in 1:L){
  w.temp <- w[-i]
  y.temp <- y[-i]
  mu.hat.i[i] <- sum(y.temp*w.temp)/sum(w.temp)
  e[i] <- y[i] - mu.hat.i[i]
  sig2.e.i <- 1/sum(w.temp) + 1/w[i]
  e.tilde.fe[i] <- e[i]/sqrt(sig2.e.i)
}

e.tilde.fe

## Compute standardized residuals of the random-effects model
mu.hat.i <- e <- e.tilde.re <- numeric(L)
for(i in 1:L){
  s2.temp <- s2[-i]
  y.temp <- y[-i]
  tau2.temp <- metapen(y.temp, s2.temp, re.method = "ML", tuning.way = "tau")$tau2.re
  w.temp <- 1/(s2.temp + tau2.temp)
  mu.hat.i[i] <- sum(y.temp*w.temp)/sum(w.temp)
  e[i] <- y[i] - mu.hat.i[i]
  var.e.i <- 1/sum(w.temp) + s2[i] + tau2.temp
  e.tilde.re[i] <- e[i]/sqrt(var.e.i)
}

e.tilde.re

## Figure 2(a) in the main content: Plot of study-specific standardized residuals of
## the meta-analysis by Bohren (2017).
pdf(file = "Figure2a.pdf", width= 2.95, height= 3.25)
```

```
metaoutliers.plot(y, s2)
text(x = e.tilde.re[4], y = (L+1) - 4, labels = "4", pos = 2, cex = 0.8)
text(x = e.tilde.re[18], y = (L+1) - 18, labels = "18", pos = 2, cex = 0.8)
dev.off()

## Exclude the 4th and 18th studies from the Bohren dataset
y.reduced <- y[-c(4, 18)]
s2.reduced <- s2[-c(4, 18)]

out.Bohren.reduced <- metagen(y.reduced, sqrt(s2.reduced), sm="OR", method.tau="ML")
print(out.Bohren.reduced)

## Compare estimates of tau
out.Bohren$tau
out.Bohren.reduced$tau

out.lam <- metapen.lambdas.cv(y, s2, n.lambda = 100, lam.c = 1.2) ## Tuning lambda
out.sd <- metapen(y, s2, re.method = "ML", tuning.way = "tau") ## Tuning tau

### Figure 3 in the main content: Four illustrative plots.
pdf(file = "Figure3.pdf", width= 4.45, height= 4.25)
par(mfrow = c(2, 2), mar = c(2.4, 2.5, 1.25, 0.5) + 0.1)

plot(log(out.lam$est[, "lambda"] + 1), out.lam$est[, "tau"], lwd = 1.5, type = "l", ann = "F",
     xlim = c(0, 8), xaxt="n", yaxt ="n")
axis(side = 1, at = 0:8, tck = -0.04, cex.axis = 0.8, mgp = c(3, 0.2, 0))
axis(side = 2, at = c(0, 0.02, 0.04, 0.06, 0.08, 0.10, 0.12, 0.14, 0.16),
     labels = c(0, 0.02, 0.04, 0.06, 0.08, 0.10, 0.12, 0.14, 0.16),
     tck = -0.04, cex.axis = 0.8, mgp = c(3, 0.4, 0))
title(xlab = expression(log(lambda + 1)), ylab = expression(hat(tau)),
      cex.lab = 0.8, line = 1.2)
title(main = "(a)", cex.main = 0.8, line = 0.8)
abline(v=log(out.lam$est[, "lambda"][which(out.lam$est[, "loss"] == min(out.lam$est[, "loss"]))] + 1),
       lwd = 1.5, lty = 2, col = "gray")
abline(h=out.lam$est[, "tau"][which(out.lam$est[, "loss"] == min(out.lam$est[, "loss"]))],
       lwd = 1.5, lty = 2, col = "gray")

plot(log(out.lam$est[, "lambda"] + 1), out.lam$est[, "loss"], xlim = c(0, 8), ylim = c(1.437, 1.492),
     lwd = 1.5, type = "l", ann = "F", xaxt = "n", yaxt = "n")
axis(side = 1, at = 0:8, tck = -0.04, cex.axis = 0.8, mgp = c(3, 0.2, 0))
axis(side = 2, at = c(1.44, 1.45, 1.46, 1.47, 1.48, 1.49),
     labels = c(1.44, 1.45, 1.46, 1.47, 1.48, 1.49),
     tck = -0.04, cex.axis = 0.8, mgp = c(3, 0.4, 0))
title(xlab = expression(log(lambda + 1)),
      ylab = expression(paste("Loss function ", ~hat(L)(lambda))),
      cex.lab = 0.8, line = 1.2)
title(main = "(b)", cex.main = 0.8, line = 0.8)
abline(v=log(out.lam$est[, "lambda"][which(out.lam$est[, "loss"] == min(out.lam$est[, "loss"]))] + 1),
       lwd = 1.5, lty = 2, col = "gray")

plot(out.lam$est[, "tau"], out.lam$est[, "loss"], lwd = 1.5, type = "l",
     xlim = c(0, 0.18), ylim = c(1.437, 1.492),
     xaxt = "n", yaxt = "n", ann = "F")
axis(side = 1, at = c(0, 0.02, 0.04, 0.06, 0.08, 0.10, 0.12, 0.14, 0.16, 0.18),
     labels = c(0, 0.02, 0.04, 0.06, 0.08, 0.10, 0.12, 0.14, 0.16, 0.18),
     tck = -0.04, cex.axis = 0.8, mgp = c(3, 0.2, 0))
axis(side = 2, at = c(1.44, 1.45, 1.46, 1.47, 1.48, 1.49),
     labels = c(1.44, 1.45, 1.46, 1.47, 1.48, 1.49),
     tck = -0.04, cex.axis = 0.8, mgp = c(3, 0.4, 0))
title(xlab = expression(hat(tau)),
      ylab = expression(paste("Loss function ", ~hat(L)(lambda))),
      cex.lab = 0.8, line = 1.2)
title(main = "(c)", cex.main = 0.8, line = 0.8)
abline(v=out.lam$est[, "tau"][which(out.lam$est[, "loss"] == min(out.lam$est[, "loss"]))],
       lwd = 1.5, lty = 2, col = "gray")

plot(out.sd$tau.cand, out.sd$loss, lwd = 1.5, type = "l",
     xlim = c(0, 0.18), ylim = c(1.448, 1.492),
     xaxt = "n", yaxt = "n", ann = "F")
axis(side = 1, at = c(0, 0.02, 0.04, 0.06, 0.08, 0.10, 0.12, 0.14, 0.16, 0.18),
     labels = c(0, 0.02, 0.04, 0.06, 0.08, 0.10, 0.12, 0.14, 0.16, 0.18),
     tck = -0.04, cex.axis = 0.8, mgp = c(3, 0.2, 0))
axis(side = 2, at = c(1.45, 1.46, 1.47, 1.48, 1.49),
     labels = c(1.45, 1.46, 1.47, 1.48, 1.49),
```

```
     tck = -0.04, cex.axis = 0.8, mgp = c(3, 0.4, 0))
title(xlab = expression(paste(tau[t])),
      ylab = expression(paste("Loss function ", ~hat(L)(tau[t]))),
      cex.lab = 0.8, line = 1.2)
title(main = "(d)", cex.main = 0.8, line = 0.8)
idx <- which(out.sd$loss == min(out.sd$loss))
abline(v = out.sd$tau.cand[idx], col = "gray", lwd = 1.5, lty = 2)

par(mfrow = c(1, 1), mar = c(5, 4, 4, 2) + 0.1)
dev.off()

## Some results of the tradeoff method by tuning lambda
## For tau, the optimal value is 0.14
out.lam$est[, "tau"][which(out.lam$est[, "loss"] == min(out.lam$est[, "loss"]))]

## For lambda, the optimal value is 33.79
out.lam$est[, "lambda"][which(out.lam$est[, "loss"] == min(out.lam$est[, "loss"]))]

## The log transfomation of the optimal lambda is 3.55
log(out.lam$est[, "lambda"][which(out.lam$est[, "loss"] == min(out.lam$est[, "loss"]))] + 1)

## Check the between-study normality assumption by p-value
## of standardized effect sizes along with their Q-Q plot
sswtest(y, s2)$pvalue
pdf(file = "FigS4.pdf", width= 2.95, height= 2.95)
par(mfrow = c(1, 1), mar=c(3, 3, .1, .2))
stdmean <- sswtest(y, s2)$stdmean
qqnorm(stdmean, main = "", xlab = "Theoretical Quantile",
       ylab = "Observed Quantile", xaxt = "n", yaxt = "n",
       ann = "F", cex = 0.6, xlim=c(-2.5, 3.2), ylim=c(-2.5, 3.2),
       bty = "l")
axis(side = 1, at = c(-2, -1, 0, 1, 2, 3), labels = c(-2, -1, 0, 1, 2, 3),
     tck=-0.04, cex.axis = 0.8, mgp=c(3, 0.5, 0))
axis(side = 2, at = c(-2, -1, 0, 1, 2, 3), labels = c(-2, -1, 0, 1, 2, 3),
     tck=-0.04, cex.axis = 0.8, mgp=c(3, 0.5, 0))
title(xlab = "Theoretical Quantile", ylab = "Observed Quantile",
      cex.lab = .8, line=1.6)
abline(0,1)
par(mfrow = c(1, 1), mar = c(5, 4, 4, 2) + 0.1)
dev.off()

## Meta-analysis by Storeb_ et al. (2015), CDSR
Storeb_ <- data.frame(
  n1 = c(65, 9, 182, 111, 133, 91, 145, 155, 53, 43, 219,
         29, 20, 151, 37, 53, 90, 87, 160, 16),
  n2 = c(71, 8, 63, 110, 46, 85, 70, 161, 47, 42, 74, 30,
         17, 152, 33, 56, 42, 90, 46, 16),
  r1 = c(5, 5, 116, 72, 71, 63, 112, 80, 40, 18, 146, 17,
         10, 70, 4, 47, 80, 14, 56, 2),
  r2 = c(4, 5, 36, 63, 38, 49, 40, 61, 27, 10, 40, 12, 5,
         64, 2, 15, 21, 12, 10, 4))
## Compute the log odds ratios and the corresponding variances
n1 <- Storeb_$n1; n2 <- Storeb_$n2; r1 <- Storeb_$r1; r2 <- Storeb_$r2; c1 <- n1 - r1; c2 <- n2 - r2
## Use Haldane-Anscombe correction to adjust the cell counts if there exist any cell equals to zero
L <- length(n1)
for(j in 1:L){
  if(r1[j]==0 | r2[j]==0 | c1[j]==0 | c2[j]==0){
    r1[j] <- r1[j] + 0.5; r2[j] <- r2[j] + 0.5; c1[j] <- c1[j] + 0.5; c2[j] <- c2[j] + 0.5}
}
rr <- (n2*r1)/(n1*r2)
Storeb_$y <- log(rr)
Storeb_$s2 <- 1/r1 - 1/n1 + 1/r2 - 1/n2
print(Storeb_)
y <- Storeb_$y; s2 <- Storeb_$s2

out.Storeb_ <- metagen(y, sqrt(s2), sm = "RR", method.tau = "ML")
print(out.Storeb_)

y_reduced <- y[-c(5, 16)]
s2_reduced <- s2[-c(5, 16)]
out.reduced <- metagen(y_reduced, sqrt(s2_reduced), sm = "RR", method.tau = "ML")
print(out.reduced)

## Figure 1(b) in the main content: Forest plot of the meta-analysis by Storeb_ et al. (2015).
```

```
cairo_pdf(filename = "Figure1b.pdf", width= 6.3, height= 3.3)
forest_reverse_log(y, s2, out = out.Storeb_, llimit = 0.35, ulimit = 4,
                   user_define_x_ticks = TRUE, x_ticks = c(0.35, 0.5, 0.71,
                   1, 1.41, 2, 3, 4), effect_type = "RR", re.method = "ML")
dev.off()

out.lam.Storeb_ <- metapen.lambdas.cv(y, s2, n.lambda = 100, lam.c = 1.2)
out.lam.Storeb_

out.sd.Storeb_ <- metapen(y, s2, re.method = "ML", tuning.way = "tau")
out.sd.Storeb_

## Compute the standardized residuals of the common-effect model
w <- 1/s2
mu.hat.i <- e <- e.tilde.fe <- numeric(L)
for(i in 1:L){
  w.temp <- w[-i]
  y.temp <- y[-i]
  mu.hat.i[i] <- sum(y.temp*w.temp)/sum(w.temp)
  e[i] <- y[i] - mu.hat.i[i]
  sig2.e.i <- 1/sum(w.temp) + 1/w[i]
  e.tilde.fe[i] <- e[i]/sqrt(sig2.e.i)
}

e.tilde.fe

## Compute standardized residuals of the random-effects model
mu.hat.i <- e <- e.tilde.re <- numeric(L)
for(i in 1:L){
  s2.temp <- s2[-i]
  y.temp <- y[-i]
  tau2.temp <- metapen(y.temp, s2.temp, re.method = "ML", tuning.way = "tau")$tau2.re
  w.temp <- 1/(s2.temp + tau2.temp)
  mu.hat.i[i] <- sum(y.temp*w.temp)/sum(w.temp)
  e[i] <- y[i] - mu.hat.i[i]
  var.e.i <- 1/sum(w.temp) + s2[i] + tau2.temp
  e.tilde.re[i] <- e[i]/sqrt(var.e.i)
}

e.tilde.re

## Figure 2(b) in the main content: Plot of study-specific standardized residuals of
## the meta-analysis by Storeb_ et al. (2015).
pdf(file = "Figure2b.pdf", width= 2.95, height= 3.25)
metaoutliers.plot(y, s2)
text(x = e.tilde.re[5], y = (L+1) - 5, labels = "5", pos = 4, cex = 0.8)
text(x = e.tilde.re[16], y = (L+1) - 16, labels = "16", pos = 2, cex = 0.8)
dev.off()

y.Storeb_<- y
s2.Storeb_ <- s2

## Check the between-study normality assumption by p-value
## of standardized effect sizes along with their Q-Q plot
sswtest(y.Storeb_, s2.Storeb_)$pvalue
pdf(file = "FigS5.pdf", width= 2.95, height= 2.95)
par(mfrow = c(1, 1), mar=c(3, 3, .1, .2))
stdmean <- sswtest(y.Storeb_, s2.Storeb_)$stdmean
qqnorm(stdmean, main = "", xlab = "Theoretical Quantile",
       ylab = "Observed Quantile", xaxt = "n", yaxt = "n",
       ann = "F", cex = 0.6, xlim=c(-2.5, 3.2), ylim=c(-2.5, 3.2),
       bty = "l")
axis(side = 1, at = c(-2, -1, 0, 1, 2, 3), labels = c(-2, -1, 0, 1, 2, 3),
     tck=-0.04, cex.axis = 0.8, mgp=c(3, 0.5, 0))
axis(side = 2, at = c(-2, -1, 0, 1, 2, 3), labels = c(-2, -1, 0, 1, 2, 3),
     tck=-0.04, cex.axis = 0.8, mgp=c(3, 0.5, 0))
title(xlab = "Theoretical Quantile", ylab = "Observed Quantile",
      cex.lab = .8, line=1.6)
abline(0,1)
par(mfrow = c(1, 1), mar = c(5, 4, 4, 2) + 0.1)
dev.off()

## Meta-analysis in Carless et al (2011), CDSR
Carless <- data.frame(
```

```
  n1 = c(128, 137, 15, 12, 20, 30, 28, 10, 25, 50, 20, 20,
         20, 64, 25, 63, 26, 24, 40, 19),
  n2 = c(128, 138, 15, 12, 20, 30, 28, 10, 29, 50, 20, 20,
         20, 52, 19, 52, 29, 28, 44, 19),
  r1 = c(59, 67, 0, 0, 1, 3, 25, 2, 23, 17, 1, 0, 0, 33,
         22, 25, 8, 8, 21, 6),
  r2 = c(86, 90, 2, 3, 14, 23, 27, 10, 26, 34, 5, 5, 5, 21,
         13, 42, 13, 10, 24, 5))
# Compute the log odds ratios and the corresponding variances
n1 <- Carless$n1; n2 <- Carless$n2; r1 <- Carless$r1; r2 <- Carless$r2; c1 <- n1 - r1; c2 <- n2 - r2
# Use Haldane-Anscombe correction to adjust the cell counts if there exist any cell equals to zero
L <- length(n1)
for(j in 1:L){
  if(r1[j]==0 | r2[j]==0 | c1[j]==0 | c2[j]==0){
    r1[j] <- r1[j] + 0.5; r2[j] <- r2[j] + 0.5; c1[j] <- c1[j] + 0.5; c2[j] <- c2[j] + 0.5}
}
odr <- (r1*c2) / (r2*c1)
Carless$y <- log(odr)
Carless$s2 <- 1/r1 + 1/c1 + 1/r2 +1/c2
print(Carless)
y <- Carless$y; s2 <- Carless$s2

out.Carless <- metagen(y, sqrt(s2), sm = "OR", method.tau = "ML")
print(out.Carless)

## Exclude the 6th and 14th studies from the Carless dataset
y.reduced <- y[-c(6, 14)]
s2.reduced <- s2[-c(6, 14)]

out.Carless.reduced <- metagen(y.reduced, sqrt(s2.reduced), sm="OR", method.tau="ML")
print(out.Carless.reduced)

## Compare estimates of tau^2
out.Carless$tau
out.Carless.reduced$tau

## Figure 1(c) in the main content: Forest plot of the meta-analysis by Carless et al. (2011).
cairo_pdf(filename = "Figure1c.pdf", width= 6.766, height= 4.078)
forest_reverse_log(y, s2, llimit = 0.002, ulimit = 8, user_define_x_ticks = TRUE,
                   x_ticks = c(0.002, 0.004, 0.008, 0.016, 0.031, 0.062, 0.125,
                   0.25, 0.5, 1, 2, 4, 8), out = out.Carless, re.method = "ML")
dev.off()

out.lam.Carl <- metapen.lambdas.cv(y, s2, n.lambda = 100, lam.c = 1.2)
out.lam.Carl

out.sd.Carl <- metapen(y, s2, re.method = "ML", tuning.way = "tau")
out.sd.Carl

## Save the effect sizes and the corresponding within-study variance of this meta-analysis
y.Carl <- y
s2.Carl <- s2

## Figure 2(c) in the main content: Plot of study-specific standardized residuals of
## the meta-analysis by Carless et al (2011).
pdf(file = "Figure2c.pdf", width= 2.95, height= 3.25)
metaoutliers.plot(y.Carl, s2.Carl, re.method = "ML")
dev.off()

## Check the between-study normality assumption by p-value
## of standardized effect sizes along with their Q-Q plot
sswtest(y.Carl, s2.Carl)$pvalue
pdf(file = "FigS6.pdf", width= 2.95, height= 2.95)
par(mfrow = c(1, 1), mar=c(3, 3, .1, .2))
stdmean <- sswtest(y.Carl, s2.Carl)$stdmean
qqnorm(stdmean, main = "", xlab = "Theoretical Quantile",
       ylab = "Observed Quantile", xaxt = "n", yaxt = "n",
       ann = "F", cex = 0.6, xlim=c(-2.5, 3.2), ylim=c(-2.5, 3.2),
       bty = "l")
axis(side = 1, at = c(-2, -1, 0, 1, 2, 3), labels = c(-2, -1, 0, 1, 2, 3),
     tck=-0.04, cex.axis = 0.8, mgp=c(3, 0.5, 0))
axis(side = 2, at = c(-2, -1, 0, 1, 2, 3), labels = c(-2, -1, 0, 1, 2, 3),
     tck=-0.04, cex.axis = 0.8, mgp=c(3, 0.5, 0))
title(xlab = "Theoretical Quantile", ylab = "Observed Quantile",
```

```
      cex.lab = .8, line=1.6)
abline(0,1)
par(mfrow = c(1, 1), mar = c(5, 4, 4, 2) + 0.1)
dev.off()

## Meta-analysis in Bjelakovic et al (2014), CDSR
Bjelakovic <- data.frame(
  n1 = c(104, 70, 2649, 80, 62, 150, 62, 95, 99, 25, 101, 195, 1634, 389,
         166, 23, 93, 41, 85, 187, 192, 313, 123, 353, 50, 51, 113,
         18176, 36, 116, 124, 1718, 121, 1762, 91, 1291, 114, 1725, 30, 42,
         177, 43, 1321, 151, 1131, 43, 34, 48, 61, 4727, 1345, 53, 39),
  n2 = c(104, 64, 2643, 80, 60, 68, 61, 97, 25, 25, 104, 196, 1636, 194, 172,
         23, 94, 41, 82, 202, 186, 312, 123, 333, 48, 53, 37,
         18106, 34, 116, 124, 1714, 122, 1955, 91, 1287, 112, 1715, 25, 70,
         171, 43, 1993, 151, 1127, 43, 35, 48, 62, 4713, 1341, 52, 81),
  r1 = c(1, 4, 836, 0, 1, 27, 0, 3, 5, 1, 16, 6, 258, 70, 25, 1, 0, 8, 1, 2,
         1, 76, 2, 2, 1, 0, 24, 744, 0, 0, 21, 15, 11, 347, 9, 282,
         1, 947, 0, 0, 11, 0, 57, 0, 40, 1, 0, 1, 7, 355, 224, 4, 0),
  r2 = c(30, 10, 917, 16, 15, 9, 5, 29, 2, 0, 23, 12, 337, 70, 33, 6, 14, 15, 28,
         32, 31, 128, 11, 26, 1, 9, 5, 1291, 4, 4, 71, 36, 8, 386, 13, 315, 15,
         1039, 9, 8, 53, 7, 131, 7, 110, 3, 3, 6, 11, 2423, 324, 4, 7))
## Compute the log odds ratios and the corresponding variances
n1 <- Bjelakovic$n1; n2 <- Bjelakovic$n2; r1 <- Bjelakovic$r1; r2 <- Bjelakovic$r2;
c1 <- n1 - r1; c2 <- n2 - r2
## Use Haldane-Anscombe correction to adjust the cell counts if there exist any cell equals to zero
L <- length(n1)
for(j in 1:L){
  if(r1[j]==0 | r2[j]==0 | c1[j]==0 | c2[j]==0){
    r1[j] <- r1[j] + 0.5; r2[j] <- r2[j] + 0.5; c1[j] <- c1[j] + 0.5; c2[j] <- c2[j] + 0.5}
}
odr <- (r1*c2) / (r2*c1)
Bjelakovic$y <- log(odr)
Bjelakovic$s2 <- 1/r1 + 1/c1 + 1/r2 +1/c2
print(Bjelakovic)
y <- Bjelakovic$y; s2 <- Bjelakovic$s2

out.Bjelakovic <- metagen(y, sqrt(s2), sm = "OR", method.tau = "ML")
print(out.Bjelakovic)

# ## Figure 1(d) in the main content: Forest plot of the meta-analysis by Bjelakovic et al. (2014).
# cairo_pdf(filename = "Figure1d.pdf", width= 6.179, height= 5.568)
# forest_reverse_log(y, s2, llimit = 0.002, ulimit = 8, user_define_x_ticks = TRUE,
#                    x_ticks = c(0.002, 0.008, 0.031, 0.125, 0.5, 1, 2, 4, 8),
#                    out = out.Bjelakovic, adjust_font_size = -0.1, re.method = "ML")
# dev.off()

out.lam.Bjela <- metapen.lambdas.cv(y, s2, n.lambda = 100, lam.c = 1.2)
out.lam.Bjela

out.sd.Bjela <- metapen(y, s2, re.method = "ML", tuning.way = "tau")
out.sd.Bjela

y.Bjela <- y
s2.Bjela <- s2

## Figure 2(d) in the main content: Plot of study-specific standardized residuals of
## the meta-analysis by Bjelakovic et al (2014).
pdf(file = "Figure2d.pdf", width= 2.95, height= 3.25)
metaoutliers.plot(y.Bjela, s2.Bjela, re.method = "ML")
dev.off()

## Figure 4 in the main content: Plots for three case studies.
pdf(file = "Figure4.pdf", width= 4.45, height= 6.195)
par(mfrow = c(3, 2), mar = c(2.4, 2.5, 1.25, 0.5) + 0.1)

plot(out.lam.Storeb_$est[, "tau"], out.lam.Storeb_$est[, "loss"], lwd = 1.5, type = "l",
     xlim = c(0, 0.27), ylim = c(1.223, 1.251),
     xaxt = "n", yaxt = "n", ann = "F")
axis(side = 1, at = c(0, 0.03, 0.06, 0.09, 0.12, 0.15, 0.18, 0.21, 0.24, 0.27),
     labels = c(0, 0.03, 0.06, 0.09, 0.12, 0.15, 0.18, 0.21, 0.24, 0.27),
     tck = -0.03, cex.axis = 0.8, mgp = c(3, 0.3, 0))
axis(side = 2, at = c(1.225, 1.23, 1.235, 1.24, 1.245, 1.25),
     labels = c(1.225, 1.23, 1.235, 1.24, 1.245, 1.25),
     tck = -0.03, cex.axis = 0.8, mgp = c(3, 0.5, 0))
```

```
title(xlab = expression(hat(tau)),
      ylab = expression(paste("Loss function ", ~hat(L)(lambda))),
      cex.lab = 0.8, line = 1.3)
title(main = "(a)", cex.main = 0.8, line = 0.5)
abline(v=out.lam.Storeb_$est[, "tau"][which(out.lam.Storeb_$est[, "loss"] ==
       min(out.lam.Storeb_$est[, "loss"]))], lwd = 1.5, lty = 2, col = "gray")

plot(out.sd.Storeb_$tau.cand, out.sd.Storeb_$loss, type = "l", lwd = 1.5,
     xlim = c(0, 0.27), ylim = c(1.2465, 1.2533),
     xaxt = "n", yaxt = "n", ann = "F")
axis(side = 1, at = c(0, 0.03, 0.06, 0.09, 0.12, 0.15, 0.18, 0.21, 0.24, 0.27),
     labels = c(0, 0.03, 0.06, 0.09, 0.12, 0.15, 0.18, 0.21, 0.24, 0.27),
     tck = -0.03, cex.axis = 0.8, mgp = c(3, 0.3, 0))
axis(side = 2, at = c(1.247, 1.248, 1.249, 1.250, 1.251, 1.252, 1.253),
     labels = c(1.247, 1.248, 1.249, 1.250, 1.251, 1.252, 1.253),
     tck = -0.03, cex.axis = 0.8, mgp = c(3, 0.5, 0))
title(xlab = expression(paste(tau[t])),
      ylab = expression(paste("Loss function ", ~hat(L)(tau[t]))),
      cex.lab = 0.8, line = 1.3)
title(main = "(b)", cex.main = 0.8, line = 0.5)
idx <- which(out.sd.Storeb_$loss == min(out.sd.Storeb_$loss))
abline(v = out.sd.Storeb_$tau.cand[idx], col = "gray", lwd = 1.5, lty = 2)

plot(out.lam.Carl$est[, "tau"], out.lam.Carl$est[, "loss"], lwd = 1.5, type = "l",
     xlim = c(0, 1), ylim = c(1.093, 1.116),
     xaxt = "n", yaxt = "n", ann = "F")
axis(side = 1, at = c(0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0),
     labels = c(0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0),
     tck = -0.03, cex.axis = 0.8, mgp = c(3, 0.3, 0))
axis(side = 2, at = c(1.095, 1.10, 1.105, 1.11, 1.115),
     labels = c(1.095, 1.10, 1.105, 1.11, 1.115),
     tck = -0.03, cex.axis = 0.8, mgp = c(3, 0.5, 0))
title(xlab = expression(hat(tau)),
      ylab = expression(paste("Loss function ", ~hat(L)(lambda))),
      cex.lab = 0.8, line = 1.3)
title(main = "(c)", cex.main = 0.8, line = 0.5)
abline(v=out.lam.Carl$est[, "tau"][which(out.lam.Carl$est[, "loss"] ==
       min(out.lam.Carl$est[, "loss"]))], lwd = 1.5, lty = 2, col = "gray")

plot(out.sd.Carl$tau.cand, out.sd.Carl$loss, type = "l", lwd = 1.5,
     xlim = c(0, 1), ylim = c(1.094, 1.111),
     xaxt = "n", yaxt = "n", ann = "F")
axis(side = 1, at = c(0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0),
     labels = c(0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0),
     tck = -0.03, cex.axis = 0.8, mgp = c(3, 0.3, 0))
axis(side = 2, at = c(1.095, 1.10, 1.105, 1.11),
     labels = c(1.095, 1.10, 1.105, 1.11),
     tck = -0.03, cex.axis = 0.8, mgp = c(3, 0.5, 0))
title(xlab = expression(paste(tau[t])),
      ylab = expression(paste("Loss function ", ~hat(L)(tau[t]))),
      cex.lab = 0.8, line = 1.3)
title(main = "(d)", cex.main = 0.8, line = 0.5)
idx <- which(out.sd.Carl$loss == min(out.sd.Carl$loss))
abline(v = out.sd.Carl$tau.cand[idx], col = "gray", lwd = 1.5, lty = 2)

plot(out.lam.Bjela$est[, "tau"], out.lam.Bjela$est[, "loss"], lwd = 1.5, type = "l",
     xlim = c(0, 0.9), ylim = c(1.038, 1.122),
     xaxt = "n", yaxt = "n", ann = "F")
axis(side = 1, at = c(0, 0.09, 0.18, 0.27, 0.36, 0.45, 0.54, 0.63, 0.72, 0.81, 0.9),
     labels = c(0, 0.09, 0.18, 0.27, 0.36, 0.45, 0.54, 0.63, 0.72, 0.81, 0.9),
     tck = -0.03, cex.axis = 0.8, mgp = c(3, 0.3, 0))
axis(side = 2, at = c(1.04, 1.05, 1.06, 1.07, 1.08, 1.09, 1.1, 1.11, 1.12),
     labels = c(1.04, 1.05, 1.06, 1.07, 1.08, 1.09, 1.10, 1.11, 1.12),
     tck = -0.03, cex.axis = 0.8, mgp = c(3, 0.5, 0))
title(xlab = expression(hat(tau)),
      ylab = expression(paste("Loss function ", ~hat(L)(lambda))),
      cex.lab = 0.8, line = 1.3)
title(main = "(e)", cex.main = 0.8, line = 0.5)
abline(v=out.lam.Bjela$est[, "tau"][which(out.lam.Bjela$est[, "loss"] ==
       min(out.lam.Bjela$est[, "loss"]))], lwd = 1.5, lty = 2, col = "gray")

plot(out.sd.Bjela$tau.cand, out.sd.Bjela$loss, type = "l", lwd = 1.5,
     xlim = c(0, 0.9), ylim = c(1.038, 1.122),
     xaxt = "n", yaxt = "n", ann = "F")
```

```
axis(side = 1, at = c(0, 0.09, 0.18, 0.27, 0.36, 0.45, 0.54, 0.63, 0.72, 0.81, 0.9),
     labels = c(0, 0.09, 0.18, 0.27, 0.36, 0.45, 0.54, 0.63, 0.72, 0.81, 0.9),
     tck = -0.03, cex.axis = 0.8, mgp = c(3, 0.3, 0))
axis(side = 2, at = c(1.04, 1.05, 1.06, 1.07, 1.08, 1.09, 1.1, 1.11, 1.12),
     labels = c(1.04, 1.05, 1.06, 1.07, 1.08, 1.09, 1.10, 1.11, 1.12),
     tck = -0.03, cex.axis = 0.8, mgp = c(3, 0.5, 0))
title(xlab = expression(paste(tau[t])),
      ylab = expression(paste("Loss function ", ~hat(L)(tau[t]))),
      cex.lab = 0.8, line = 1.3)
title(main = "(f)", cex.main = 0.8, line = 0.5)
idx <- which(out.sd.Bjela$loss == min(out.sd.Bjela$loss))
abline(v = out.sd.Bjela$tau.cand[idx], col = "gray", lwd = 1.5, lty = 2)

par(mfrow = c(1, 1), mar = c(5, 4, 4, 2) + 0.1)
dev.off()

## Check the between-study normality assumption by p-value
## of standardized effect sizes along with their Q-Q plot
sswtest(y.Bjela, s2.Bjela)$pvalue
pdf(file = "FigS7.pdf", width= 2.95, height= 2.95)
par(mfrow = c(1, 1), mar=c(3, 3, .1, .2))
stdmean <- sswtest(y.Bjela, s2.Bjela)$stdmean
qqnorm(stdmean, main = "", xlab = "Theoretical Quantile",
       ylab = "Observed Quantile", xaxt = "n", yaxt = "n",
       ann = "F", cex = 0.6, xlim=c(-2.5, 3.2), ylim=c(-2.5, 3.2),
       bty = "l")
axis(side = 1, at = c(-2, -1, 0, 1, 2, 3), labels = c(-2, -1, 0, 1, 2, 3),
     tck=-0.04, cex.axis = 0.8, mgp=c(3, 0.5, 0))
axis(side = 2, at = c(-2, -1, 0, 1, 2, 3), labels = c(-2, -1, 0, 1, 2, 3),
     tck=-0.04, cex.axis = 0.8, mgp=c(3, 0.5, 0))
title(xlab = "Theoretical Quantile", ylab = "Observed Quantile",
      cex.lab = .8, line=1.6)
abline(0,1)
par(mfrow = c(1, 1), mar = c(5, 4, 4, 2) + 0.1)
dev.off()

## Meta-analysis in Keus et al. (2006), CDSR
Keus <- data.frame(
  n1 = c(29, 14, 15, 100, 20, 15, 35, 40, 50, 30, 16, 6, 10, 30, 25,
         15, 50, 45, 7, 7, 20, 50, 10, 10, 35, 58, 58),
  n2 = c(21, 14, 12, 100, 20, 15, 35, 38, 50, 27, 16, 6, 5, 28, 22, 12,
         51, 42, 7, 7, 20, 50, 10, 10, 35, 60, 52),
  r1 = c(1, 0, 1, 13, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 1, 0,
         5, 0, 1, 0, 1, 0, 0, 0, 14, 0, 0),
  r2 = c(4, 0, 0, 16, 1, 1, 10, 0, 0, 1, 0, 0, 1, 6, 2,
         3, 1, 0, 0, 0, 2, 2, 0, 0, 8, 5, 14))
# Compute the log odds ratios and the corresponding variances
n1 <- Keus$n1; n2 <- Keus$n2; r1 <- Keus$r1; r2 <- Keus$r2; c1 <- n1 - r1; c2 <- n2 - r2
# Use Haldane-Anscombe correction to adjust the cell counts if there exist any cell equals to zero
L <- length(n1)
for(j in 1:L){
  if(r1[j]==0 | r2[j]==0 | c1[j]==0 | c2[j]==0){
    r1[j] <- r1[j] + 0.5; r2[j] <- r2[j] + 0.5; c1[j] <- c1[j] + 0.5; c2[j] <- c2[j] + 0.5}
}
odr <- (r1*c2) / (r2*c1)
Keus$y <- log(odr)
Keus$s2 <- 1/r1 + 1/c1 + 1/r2 +1/c2
print(Keus)
y <- Keus$y; s2 <- Keus$s2

out.Keus <- metagen(y, sqrt(s2), sm = "OR", method.tau = "ML")
print(out.Keus)

## Figure S1 in the Supplementary Materials: Forest plot of the meta-analysis by Keus et al. (2006)
cairo_pdf(filename = "FigS1.pdf", width= 6.328, height= 4.510)
forest_reverse_log(y, s2, out = out.Keus, adjust_font_size = -0.05,
                   llimit = 0.015, ulimit = 12, user_define_x_ticks = TRUE,
                   x_ticks = c(0.015, 0.03, 0.06, 0.125, 0.25, 0.5, 1, 2,
                   4, 8, 12), re.method = "ML")
dev.off()

out.lam.Keus <- metapen.lambdas.cv(y, s2, n.lambda = 100, lam.c = 1.2)
out.lam.Keus
```

```
out.sd.Keus <- metapen(y, s2, re.method = "ML", tuning.way = "tau")
out.sd.Keus

y.Keus <- y
s2.Keus <- s2

## Meta-analysis in Adams et al. (2018), CDSR
Adams <- data.frame(
  y = c(4.30, 7.50, 5.15, 5.00, 4.80, 4.00, 0.90, 3.50, 7.95, 6.65,
        7.50, 13.00, 6.20, 8.30, 13.60, 7.80, 5.00, 4.90, 5.20, 5.00,
        3.65, 5.05, 1.50, 5.80, 6.60, 7.70, 3.80, 11.00, 2.20, 32.00,
        5.15, 7.70),
  s2 = c(0.3349, 1.8824, 1.3741, 1.9600, 0.1231, 14.2219, 21.3333,
         0.7118, 4.2630, 7.3143, 13.4762, 16.0000, 0.8127, 8.5334,
         42.6670, 8.5334, 1.6900, 0.8100, 11.4190, 1.9600, 4.0634,
         0.5441, 3.8789, 3.6573, 10.2400, 4.1290, 3.5502, 10.2400,
         0.1555, 42.6670, 7.2900, 6.5644))
print(Adams)
y <- Adams$y; s2 <- Adams$s2

out.Adams <- metagen(y, sqrt(s2), sm = "MD", method.tau = "ML")
print(out.Adams)

y_reduced <- y[-c(29, 30)]
s2_reduced <- s2[-c(29, 30)]
out.reduced <- metagen(y_reduced, sqrt(s2_reduced), sm = "MD", method.tau = "ML")
print(out.reduced)

## Figure S2 in the Supplementary Materials: Forest plot of the meta-analysis by Adams et al. (2018).
cairo_pdf(filename = "FigS2.pdf", width= 6.297, height= 3.906)
forest_original_effects(y, s2, out = out.Adams, adjust_font_size = -0.04,
                        llimit = -6.5, ulimit = 36, user_define_x_ticks = TRUE,
                        x_ticks = c(-6.5, -3, 0, 3, 6.5, 10.5, 16, 21, 26, 31, 36),
                        re.method = "ML")
dev.off()

out.lam.Adams <- metapen.lambdas.cv(y, s2, n.lambda = 100, lam.c = 1.2)
out.lam.Adams

out.sd.Adams <- metapen(y, s2, re.method = "ML", tuning.way = "tau")
out.sd.Adams

y.Adams<- y
s2.Adams <- s2

## Figure S3 in the Supplementary Materials: Plots for two additional meta-analyses.
pdf(file = "FigS3.pdf", width= 5.656, height= 4.024)
par(mfrow = c(2, 3), mar = c(2.4, 2.8, 1, 1) + 0.1)

plot(out.lam.Keus$est[, "tau"], out.lam.Keus$est[, "loss"], lwd = 1.5, type = "l",
     xlim = c(0, 0.7), ylim = c(0.9545, 0.9677), xaxt = "n", yaxt = "n", ann = "F")
axis(side = 1, at = c(0, 0.07, 0.14, 0.21, 0.28, 0.35, 0.42, 0.49, 0.56, 0.63, 0.7),
     labels = c(0, 0.07, 0.14, 0.21, 0.28, 0.35, 0.42, 0.49, 0.56, 0.63, 0.7),
     tck = -0.03, cex.axis = 0.8, mgp = c(3, 0.3, 0))
axis(side = 2, at = c(0.955, 0.9575, 0.96, 0.9625, 0.965, 0.9675), labels = c(0.955,
     0.9575, 0.96, 0.9625, 0.965, 0.9675),
     tck = -0.03, cex.axis = 0.8, mgp = c(3, 0.5, 0))
title(xlab = expression(hat(tau)),
      ylab = expression(paste("Loss function ", ~hat(L)(lambda))),
      cex.lab = 0.8, line = 1.3)
title(main = "(a)", cex.main = 0.8, line = 0.5)
abline(v=out.lam.Keus$est[, "tau"][which(out.lam.Keus$est[, "loss"] ==
       min(out.lam.Keus$est[, "loss"]))], lwd = 1.5, lty = 2, col = "gray")

plot(out.sd.Keus$tau.cand, out.sd.Keus$loss, type = "l", lwd = 1.5,
     xlim = c(0, 0.7), ylim = c(0.9608, 0.9652),
     xaxt = "n", yaxt = "n", ann = "F")
axis(side = 1, at = c(0, 0.07, 0.14, 0.21, 0.28, 0.35, 0.42, 0.49, 0.56, 0.63, 0.7),
     labels = c(0, 0.07, 0.14, 0.21, 0.28, 0.35, 0.42, 0.49, 0.56, 0.63, 0.7),
     tck = -0.03, cex.axis = 0.8, mgp = c(3, 0.3, 0))
axis(side = 2, at = c(0.961, 0.962, 0.963, 0.964, 0.965), labels = c(0.961,
     0.962, 0.963, 0.964, 0.965),tck = -0.03, cex.axis = 0.8, mgp = c(3, 0.5, 0))
title(xlab = expression(paste(tau[t])),
      ylab = expression(paste("Loss function ", ~hat(L)(tau[t]))),
```

```
      cex.lab = 0.8, line = 1.3)
title(main = "(b)", cex.main = 0.8, line = 0.5)
idx <- which(out.sd.Keus$loss == min(out.sd.Keus$loss))
abline(v = out.sd.Keus$tau.cand[idx], col = "gray", lwd = 1.5, lty = 2)

metaoutliers.plot(y.Keus, s2.Keus)
title(main = "(c)", cex.main = 0.8, line = 0.5)

plot(out.lam.Adams$est[, "tau"], out.lam.Adams$est[, "loss"], lwd = 1.5, type = "l",
     xlim = c(0, 1.2), ylim = c(1.568, 1.632),
     xaxt = "n", yaxt = "n", ann = "F")
axis(side = 1, at = c(0, 0.12, 0.24, 0.36, 0.48, 0.60, 0.72, 0.84, 0.96, 1.08, 1.2),
     labels = c(0, 0.12, 0.24, 0.36, 0.48, 0.60, 0.72, 0.84, 0.96, 1.08, 1.2),
     tck = -0.03, cex.axis = 0.8, mgp = c(3, 0.3, 0))
axis(side = 2, at = c(1.57, 1.58, 1.59, 1.60, 1.61, 1.62, 1.63),
     labels = c(1.57, 1.58, 1.59, 1.60, 1.61, 1.62, 1.63),
     tck = -0.03, cex.axis = 0.8, mgp = c(3, 0.5, 0))
title(xlab = expression(hat(tau)),
      ylab = expression(paste("Loss function ", ~hat(L)(lambda))),
      cex.lab = 0.8, line = 1.3)
title(main = "(d)", cex.main = 0.8, line = 0.5)
abline(v=out.lam.Adams$est[, "tau"][which(out.lam.Adams$est[, "loss"] ==
       min(out.lam.Adams$est[, "loss"]))], lwd = 1.5, lty = 2, col = "gray")

plot(out.sd.Adams$tau.cand, out.sd.Adams$loss, type = "l", lwd = 1.5,
     xlim = c(0, 1.2), ylim = c(1.6255, 1.636),
     xaxt = "n", yaxt = "n", ann = "F")
axis(side = 1, at = c(0, 0.12, 0.24, 0.36, 0.48, 0.60, 0.72, 0.84, 0.96, 1.08, 1.2),
     labels = c(0, 0.12, 0.24, 0.36, 0.48, 0.60, 0.72, 0.84, 0.96, 1.08, 1.2),
     tck = -0.03, cex.axis = 0.8, mgp = c(3, 0.3, 0))
axis(side = 2, at = c(1.626, 1.627, 1.628, 1.629, 1.630, 1.631, 1.632, 1.633,
     1.634, 1.635, 1.636), labels = c(1.626, 1.627, 1.628, 1.629, 1.630, 1.631,
     1.632, 1.633, 1.634, 1.635, 1.636), tck = -0.03, cex.axis = 0.8, mgp = c(3, 0.5, 0))
title(xlab = expression(paste(tau[t])),
      ylab = expression(paste("Loss function ", ~hat(L)(tau[t]))),
      cex.lab = 0.8, line = 1.3)
title(main = "(e)", cex.main = 0.8, line = 0.5)
idx <- which(out.sd.Adams$loss == min(out.sd.Adams$loss))
abline(v = out.sd.Adams$tau.cand[idx], col = "gray", lwd = 1.5, lty = 2)

metaoutliers.plot(y.Adams, s2.Adams)
title(main = "(f)", cex.main = 0.8, line = 0.5)

par(mfrow = c(1, 1), mar = c(5, 4, 4, 2) + 0.1)
dev.off()



##############################
#### Simulated data analysis
##############################

## dat.outlier() is used to generate a meta-analysis with outlying studies.
## arguments: n is the total number of studies;
##    n.outlier is the number of outlying studies;
##    mu represents the underlying true overall effect size;
##    C is a discrepancy added to studies to make them outlying;
##    s.low and s.upp are the lower and upper bounds of the uniform
##    distribution which is used to simulate the within-study standard deviation;
##    tau is the true between-study standard deivation.
dat.outlier <- function(n = 30, n.outlier = 5, mu = 0, C = 3, s.low = 0.5, s.upp = 1, tau = 1){
  mui <- rnorm(n, mean = mu, sd = tau)
  if(n.outlier > 0) mui[1:n.outlier] <- mui[1:n.outlier] + C
  s <- runif(n, min = s.low, max = s.upp)
  y <- rnorm(n, mean = mui, sd = s)
  out <- list(y = y, s2 = s^2)
  return(out)
}

## Implement the simulation study to compare the performance of
## the penalization method by tuning lambda with that of two conventional models.
n.sim <- 5000
n <- 30
mu <- 0
```

```
s.low <- 0.1
s.upp <- 1

begin <- Sys.time()
for(n.outlier in 0:6){
  taus <- c(0, 0.5, 1, 2)
  for(tau in taus){
    C <- 3*sqrt(s.upp^2 + tau^2)
    set.seed(1234)
    mu.fe <- mu.re <- mu.opt <- matrix(NA, n.sim, 3)
    for(i in 1:n.sim){
      dat <- dat.outlier(n, n.outlier, mu, C, s.low, s.upp, tau)
      out <- metapen(dat$y, dat$s2, re.method = "ML", tuning.way = "lambda")
      mu.fe[i, 1] <- out$mu.fe
      mu.fe[i, 2] <- out$mu.fe - 1.96*out$se.fe
      mu.fe[i, 3] <- out$mu.fe + 1.96*out$se.fe
      mu.re[i, 1] <- out$mu.re
      mu.re[i, 2] <- out$mu.re - 1.96*out$se.re
      mu.re[i, 3] <- out$mu.re + 1.96*out$se.re
      mu.opt[i, 1] <- out$mu.opt
      mu.opt[i, 2] <- out$mu.opt - 1.96*out$se.opt
      mu.opt[i, 3] <- out$mu.opt + 1.96*out$se.opt
    }
    bias.fe <- mean(mu.fe[,1]) - mu
    mse.fe <- mean((mu.fe[,1] - mu)^2)
    covprob.fe <- mean(mu.fe[,2] < mu & mu.fe[,3] > mu)
    bias.re <- mean(mu.re[,1]) - mu
    mse.re <- mean((mu.re[,1] - mu)^2)
    covprob.re <- mean(mu.re[,2] < mu & mu.re[,3] > mu)
    bias.opt <- mean(mu.opt[,1]) - mu
    mse.opt <- mean((mu.opt[,1] - mu)^2)
    covprob.opt <- mean(mu.opt[,2] < mu & mu.opt[,3] > mu)
    rst <- rbind(bias.fe, mse.fe, covprob.fe,
                 bias.re, mse.re, covprob.re,
                 bias.opt, mse.opt, covprob.opt)
    write.table(rst, paste("Tune_lamb_", "tau_", tau,
                           "_n.outlier_", n.outlier, ".txt", sep = ""),
                row.names = rownames(rst), col.names = FALSE)
  }
}
end <- Sys.time()
end - begin
## Time difference of 13.96 days

## Implement the simulation study to compare the performance of
## the penalization method by tuning tau with that of two conventional models.
begin <- Sys.time()
for(n.outlier in 0:6){
  taus <- c(0, 0.5, 1, 2)
  for(tau in taus){
    C <- 3*sqrt(s.upp^2 + tau^2)
    set.seed(1234)
    mu.fe <- mu.re <- mu.opt <- matrix(NA, n.sim, 3)
    for(i in 1:n.sim){
      dat <- dat.outlier(n, n.outlier, mu, C, s.low, s.upp, tau)
      out <- metapen(dat$y, dat$s2, re.method = "ML", tuning.way = "tau")
      mu.fe[i, 1] <- out$mu.fe
      mu.fe[i, 2] <- out$mu.fe - 1.96*out$se.fe
      mu.fe[i, 3] <- out$mu.fe + 1.96*out$se.fe
      mu.re[i, 1] <- out$mu.re
      mu.re[i, 2] <- out$mu.re - 1.96*out$se.re
      mu.re[i, 3] <- out$mu.re + 1.96*out$se.re
      mu.opt[i, 1] <- out$mu.opt
      mu.opt[i, 2] <- out$mu.opt - 1.96*out$se.opt
      mu.opt[i, 3] <- out$mu.opt + 1.96*out$se.opt
    }
    bias.fe <- mean(mu.fe[,1]) - mu
    mse.fe <- mean((mu.fe[,1] - mu)^2)
    covprob.fe <- mean(mu.fe[,2] < mu & mu.fe[,3] > mu)
    bias.re <- mean(mu.re[,1]) - mu
    mse.re <- mean((mu.re[,1] - mu)^2)
    covprob.re <- mean(mu.re[,2] < mu & mu.re[,3] > mu)
    bias.opt <- mean(mu.opt[,1]) - mu
    mse.opt <- mean((mu.opt[,1] - mu)^2)
```

```
    covprob.opt <- mean(mu.opt[,2] < mu & mu.opt[,3] > mu)
    rst <- rbind(bias.fe, mse.fe, covprob.fe,
                 bias.re, mse.re, covprob.re,
                 bias.opt, mse.opt, covprob.opt)
    write.table(rst, paste("Tune_tau_", "tau_", tau,
                           "_n.outlier_", n.outlier, ".txt", sep = ""),
                row.names = rownames(rst), col.names = FALSE)
  }
}
end <- Sys.time()
end - begin
## Time difference of 3.73 hours

## dat.lowerI2() is used to generate a meta-analysis with lower I2 values
## (approximately) by controling the value of tau.
dat.lowerI2 <- function(n = 25, mu = 0, s.low = 0.5, s.upp = 1, tau = 0.55){
  mui <- rnorm(n, mean = mu, sd = tau)
  s <- runif(n, min = s.low, max = s.upp)
  y <- rnorm(n, mean = mui, sd = s)
  out <- list(y = y, s2 = s^2)
  return(out)
}

## Reset the simulation settings
n.sim <- 5000
ns <- c(5, 10, 25)
mu <- 0
s.low <- 0.1
s.upp <- 1

## Compare the performance of the penalization method by tuning lambda
## with the CE and RE models when there are no outliers and I2
## approximately equal to 0.15, 0.25, 0.5.
begin <- Sys.time()
for(n in ns){
  taus <- c(0.23, 0.32, 0.55)
  for(tau in taus){
    set.seed(1234)
    mu.fe <- mu.re <- mu.opt <- matrix(NA, n.sim, 3)
    for(i in 1:n.sim){
      dat <- dat.lowerI2(n, mu, s.low, s.upp, tau)
      out <- metapen(dat$y, dat$s2, re.method = "ML", tuning.way = "lambda")
      mu.fe[i, 1] <- out$mu.fe
      mu.fe[i, 2] <- out$mu.fe - 1.96*out$se.fe
      mu.fe[i, 3] <- out$mu.fe + 1.96*out$se.fe
      mu.re[i, 1] <- out$mu.re
      mu.re[i, 2] <- out$mu.re - 1.96*out$se.re
      mu.re[i, 3] <- out$mu.re + 1.96*out$se.re
      mu.opt[i, 1] <- out$mu.opt
      mu.opt[i, 2] <- out$mu.opt - 1.96*out$se.opt
      mu.opt[i, 3] <- out$mu.opt + 1.96*out$se.opt
    }
    bias.fe <- mean(mu.fe[,1]) - mu
    mse.fe <- mean((mu.fe[,1] - mu)^2)
    covprob.fe <- mean(mu.fe[,2] < mu & mu.fe[,3] > mu)
    bias.re <- mean(mu.re[,1]) - mu
    mse.re <- mean((mu.re[,1] - mu)^2)
    covprob.re <- mean(mu.re[,2] < mu & mu.re[,3] > mu)
    bias.opt <- mean(mu.opt[,1]) - mu
    mse.opt <- mean((mu.opt[,1] - mu)^2)
    covprob.opt <- mean(mu.opt[,2] < mu & mu.opt[,3] > mu)
    rst <- rbind(bias.fe, mse.fe, covprob.fe,
                 bias.re, mse.re, covprob.re,
                 bias.opt, mse.opt, covprob.opt)
    write.table(rst, paste("Tune_lamb_", "tau_", tau,
                           "_n_", n, ".txt", sep = ""),
                row.names = rownames(rst), col.names = FALSE)
  }
}
end <- Sys.time()
end - begin
## Time difference of 1.64 days

## Compare the performance of the penalization method by tuning tau
```

```
## with the CE and RE models when there are no outliers and I2
## approximately equal to 0.15, 0.25, 0.5.
begin <- Sys.time()
for(n in ns){
  taus <- c(0.23, 0.32, 0.55)
  for(tau in taus){
    set.seed(1234)
    mu.fe <- mu.re <- mu.opt <- matrix(NA, n.sim, 3)
    for(i in 1:n.sim){
      dat <- dat.lowerI2(n, mu, s.low, s.upp, tau)
      out <- metapen(dat$y, dat$s2, re.method = "ML", tuning.way = "tau")
      mu.fe[i, 1] <- out$mu.fe
      mu.fe[i, 2] <- out$mu.fe - 1.96*out$se.fe
      mu.fe[i, 3] <- out$mu.fe + 1.96*out$se.fe
      mu.re[i, 1] <- out$mu.re
      mu.re[i, 2] <- out$mu.re - 1.96*out$se.re
      mu.re[i, 3] <- out$mu.re + 1.96*out$se.re
      mu.opt[i, 1] <- out$mu.opt
      mu.opt[i, 2] <- out$mu.opt - 1.96*out$se.opt
      mu.opt[i, 3] <- out$mu.opt + 1.96*out$se.opt
    }
    bias.fe <- mean(mu.fe[,1]) - mu
    mse.fe <- mean((mu.fe[,1] - mu)^2)
    covprob.fe <- mean(mu.fe[,2] < mu & mu.fe[,3] > mu)
    bias.re <- mean(mu.re[,1]) - mu
    mse.re <- mean((mu.re[,1] - mu)^2)
    covprob.re <- mean(mu.re[,2] < mu & mu.re[,3] > mu)
    bias.opt <- mean(mu.opt[,1]) - mu
    mse.opt <- mean((mu.opt[,1] - mu)^2)
    covprob.opt <- mean(mu.opt[,2] < mu & mu.opt[,3] > mu)
    rst <- rbind(bias.fe, mse.fe, covprob.fe,
                 bias.re, mse.re, covprob.re,
                 bias.opt, mse.opt, covprob.opt)
    write.table(rst, paste("Tune_tau_", "tau_", tau,
                           "_n_", n, ".txt", sep = ""),
                row.names = rownames(rst), col.names = FALSE)
  }
}
end <- Sys.time()
end - begin
## Time difference of 44.17 mins

## dat.simt() is used to generate a meta-analysis which have
## t-distributed random effects (note that the t-distribution is heavy tail).
## arguments are very similar to dat.outlier() except df represents the
##    degree of freedoms of the t-distribution
dat.simt <- function(n = 30, mu = 0, s.low = 0.5, s.upp = 1, tau = 1, df = 5){
  mui <- mu + tau * rt(n, df)
  s <- runif(n, min = s.low, max = s.upp)
  y <- rnorm(n, mean = mui, sd = s)
  out <- list(y = y, s2 = s^2)
  return(out)
}

## When the between-study variances are t-distributed, implementing the simulation
## study to compare the performance of the penalization method by tuning lambda
## with that of two conventional models.
n.sim <- 5000
n <- 30
mu <- 0
s.low <- 0.1
s.upp <- 1
df <- 5  ## df should greater than 2
taus <- c(0, 0.21, 0.43, 0.85)

begin <- Sys.time()
for(tau in taus){
  set.seed(1234)
  mu.fe <- mu.re <- mu.opt <- matrix(NA, n.sim, 3)
  for(i in 1:n.sim){
    dat <- dat.simt(n, mu, s.low, s.upp, tau, df)
    out <- metapen(dat$y, dat$s2, re.method = "ML", tuning.way = "lambda")
    mu.fe[i, 1] <- out$mu.fe
    mu.fe[i, 2] <- out$mu.fe - 1.96*out$se.fe
```

```
    mu.fe[i, 3] <- out$mu.fe + 1.96*out$se.fe
    mu.re[i, 1] <- out$mu.re
    mu.re[i, 2] <- out$mu.re - 1.96*out$se.re
    mu.re[i, 3] <- out$mu.re + 1.96*out$se.re
    mu.opt[i, 1] <- out$mu.opt
    mu.opt[i, 2] <- out$mu.opt - 1.96*out$se.opt
    mu.opt[i, 3] <- out$mu.opt + 1.96*out$se.opt
  }
  bias.fe <- mean(mu.fe[,1]) - mu
  mse.fe <- mean((mu.fe[,1] - mu)^2)
  covprob.fe <- mean(mu.fe[,2] < mu & mu.fe[,3] > mu)
  bias.re <- mean(mu.re[,1]) - mu
  mse.re <- mean((mu.re[,1] - mu)^2)
  covprob.re <- mean(mu.re[,2] < mu & mu.re[,3] > mu)
  bias.opt <- mean(mu.opt[,1]) - mu
  mse.opt <- mean((mu.opt[,1] - mu)^2)
  covprob.opt <- mean(mu.opt[,2] < mu & mu.opt[,3] > mu)
  rst <- rbind(bias.fe, mse.fe, covprob.fe,
               bias.re, mse.re, covprob.re,
               bias.opt, mse.opt, covprob.opt)
  write.table(rst, paste("tRE_Tunelamb_", "tau_", tau, ".txt", sep = ""),
              row.names = rownames(rst), col.names = FALSE)
}
end <- Sys.time()
end - begin
## Time difference of 1.41 days

## When the between-study variances are t-distributed, implementing the simulation
## study to compare the performance of the penalization method by tuning tau
## with that of two conventional models.
begin <- Sys.time()
for(tau in taus){
  set.seed(1234)
  mu.fe <- mu.re <- mu.opt <- matrix(NA, n.sim, 3)
  for(i in 1:n.sim){
    dat <- dat.simt(n, mu, s.low, s.upp, tau, df)
    out <- metapen(dat$y, dat$s2, re.method = "ML", tuning.way = "tau")
    mu.fe[i, 1] <- out$mu.fe
    mu.fe[i, 2] <- out$mu.fe - 1.96*out$se.fe
    mu.fe[i, 3] <- out$mu.fe + 1.96*out$se.fe
    mu.re[i, 1] <- out$mu.re
    mu.re[i, 2] <- out$mu.re - 1.96*out$se.re
    mu.re[i, 3] <- out$mu.re + 1.96*out$se.re
    mu.opt[i, 1] <- out$mu.opt
    mu.opt[i, 2] <- out$mu.opt - 1.96*out$se.opt
    mu.opt[i, 3] <- out$mu.opt + 1.96*out$se.opt
  }
  bias.fe <- mean(mu.fe[,1]) - mu
  mse.fe <- mean((mu.fe[,1] - mu)^2)
  covprob.fe <- mean(mu.fe[,2] < mu & mu.fe[,3] > mu)
  bias.re <- mean(mu.re[,1]) - mu
  mse.re <- mean((mu.re[,1] - mu)^2)
  covprob.re <- mean(mu.re[,2] < mu & mu.re[,3] > mu)
  bias.opt <- mean(mu.opt[,1]) - mu
  mse.opt <- mean((mu.opt[,1] - mu)^2)
  covprob.opt <- mean(mu.opt[,2] < mu & mu.opt[,3] > mu)
  rst <- rbind(bias.fe, mse.fe, covprob.fe,
               bias.re, mse.re, covprob.re,
               bias.opt, mse.opt, covprob.opt)
  write.table(rst, paste("tRE_Tunetau_", "tau_", tau, ".txt", sep = ""),
              row.names = rownames(rst), col.names = FALSE)
}
end <- Sys.time()
end - begin
## Time difference of 29.68 mins
```

References

1. Wang C.C., Lee W.C. Evaluation of the normality assumption in meta-analyses. *American Journal of Epidemiology.* 2020;189(3):235–242.

2. Bohren M.A., Hofmeyr G.J., Sakala C., Fukuzawa R.K., Cuthbert A. Continuous support for women during childbirth. *Cochrane Database of Systematic Reviews.* 2017;7:Art. No.: CD003766.

3. Storebø O.J., Ramstad E., Krogh H.B., et al. Methylphenidate for children and adolescents with attention deficit hyperactivity disorder (ADHD). *Cochrane Database of Systematic Reviews.* 2015;11:Art. No.: CD009885.

4. Carless P.A., Rubens F.D., Anthony D.M., O'Connell D., Henry D.A. Platelet-rich-plasmapheresis for minimising peri-operative allogeneic blood transfusion. *Cochrane Database of Systematic Reviews.* 2011;3:Art. No.: CD004172.

5. Bjelakovic G., Gluud L.L., Nikolova D., et al. Vitamin D supplementation for prevention of mortality in adults. *Cochrane Database of Systematic Reviews.* 2014;1:Art. No.: CD007470.

6. Keus F., de Jong J., Gooszen H.G., Laarhoven C.J.H.M. Laparoscopic versus open cholecystectomy for patients with symptomatic cholecystolithiasis. *Cochrane Database of Systematic Reviews.* 2006;4:Art. No.: CD006231.

7. Adams S.P., Sekhon S.S., Tsang M., Wright J.M. Fluvastatin for lowering lipids. *Cochrane Database of Systematic Reviews.* 2018;3:Art. No.: CD012282.