Package 'altmeta'

October 13, 2025

Type Package

Title Alternative Meta-Analysis Methods

Version 4.3

Date 2025-10-12

Maintainer Lifeng Lin <lifenglin@arizona.edu>

Depends R (>= 3.5.0)

Imports rjags (>= 4-6), coda, graphics, grDevices, lme4, Matrix, metafor (>= 3.0-2), methods, stats, utils

SystemRequirements JAGS 4.x.y (http://mcmc-jags.sourceforge.net)

Description Provides alternative statistical methods for meta-analysis, including:
- bivariate generalized linear mixed models for synthesizing odds ratios, relative risks, and risk differences (Chu et al., 2012 <doi:10.1177/0962280210393712>)
- tests and measures for between-study heterogeneity (Lin et al., 2017 <doi:10.1111/biom.12543>; Wang et al., 2022 <doi:10.1002/sim.9261>);
- measures, tests, and visualization tools for publication bias or small-study effects (Lin and Chu, 2018 <doi:10.1111/biom.12817>; Lin, 2019 <doi:10.1002/jrsm.1340>; Lin, 2020 <doi:10.1177/0962280220910172>; Shi et al., 2020 <doi:10.1002/jrsm.1415>);
- meta-analysis of combining standardized mean differences and odds ratios (Jing et al., 2023 <doi:10.1080/10543406.2022.2105345>);
- meta-analysis of diagnostic tests for synthesizing sensitivities, specificities, etc. (Reitsma et al., 2005 <doi:10.1016/j.jclinepi.2005.02.022>; Chu and Cole, 2006 <doi:10.1016/j.jclinepi.2006.06.011>);
- meta-analysis methods for synthesizing proportions (Lin and Chu, 2020 <doi:10.1097/ede.0000000000001232>);
- models for multivariate meta-analysis, measures of inconsistency degrees of freedom in Bayesian network meta-analysis, and predictive P-score (Lin and Chu, 2018 <doi:10.1002/jrsm.1293>; Lin, 2020 <doi:10.1080/10543406.2020.1852247>; Rosenberger et al., 2021 <doi:10.1186/s12874-021-01397-5>).

License GPL (>= 2)

NeedsCompilation no

Author Lifeng Lin [aut, cre] (ORCID: <https://orcid.org/0000-0002-3562-9816>),
Yaqi Jing [ctb],
Kristine J. Rosenberger [ctb],
Linyu Shi [ctb],
Yipeng Wang [ctb],
Xing Xing [ctb] (ORCID: <https://orcid.org/0000-0001-9394-8957>),
Zhiyuan Yu [ctb],
Haitao Chu [aut] (ORCID: <https://orcid.org/0000-0003-0932-598X>)

Repository CRAN

Date/Publication 2025-10-13 20:00:02 UTC

Contents

`dat.adams` *A Meta-Analysis on the Effects of Fluvastatin 20 mg/day on High-Density Lipoprotein*

Description

This meta-analysis serves as an example to illustrate the usage of the function `meta.pen`.

Usage

```
data("dat.ha")
```

Format

A data frame containing 32 studies.

`y` the observed effect size (mean difference) for each study in the meta-analysis.

`s2` the within-study variance for each study.

Source

Adams SP, Sekhon SS, Tsang M, Wright JM (2018). "Fluvastatin for lowering lipids." *Cochrane Database of Systematic Reviews*, **7**. Art. No.: CD012282. <doi:10.1002/14651858.CD012282.pub2>

dat.aex — *A Meta-Analysis for Evaluating the Effect of Aerobic Exercise on Visceral Adipose Tissue Content/Volume*

Description

This meta-analysis serves as an example to illustrate function usage in the package **altmeta**.

Usage

```
data("dat.aex")
```

Format

A data frame containing 29 studies with the observed effect sizes and their within-study variances.

`y` the observed effect size for each collected study in the meta-analysis.

`s2` the within-study variance for each study.

Source

Ismail I, Keating SE, Baker MK, Johnson NA (2012). "A systematic review and meta-analysis of the effect of aerobic vs. resistance exercise training on visceral fat." *Obesity Reviews*, **13**(1), 68–91. <doi:10.1111/j.1467789X.2011.00931.x>

dat.annane — *A Meta-Analysis for Comparing the Effect of Steroids vs. Control in the Length of Intensive Care Unit (ICU) Stay*

Description

This dataset serves as an example of meta-analysis of mean differences.

Usage

```
data("dat.annane")
```

Format

A data frame with 12 studies with the following 5 variables within each study.

`y` point esimates of mean differences.

`s2` sample variances of mean differences.

`n1` sample sizes in treatment group 1 (steroids).

`n2` sample sizes in treatment group 2 (control).

`n` total sample sizes.

Source

Annane D, Bellissant E, Bollaert PE, Briegel J, Keh D, Kupfer Y (2015). "Corticosteroids for treating sepsis." *Cochrane Database of Systematic Reviews*, **12**, Art. No.: CD002243. <doi:10.1002/14651858.CD002243.pub3>

`dat.baker` *A Network Meta-Analysis on Effects of Pharmacologic Treatments for Chronic Obstructive Pulmonary Disease*

Description

This dataset serves as an example of network meta-analysis with binary outcomes.

Usage

```
data("dat.baker")
```

Format

A dataset of network meta-analysis with binary outcomes, containing 38 studies and 5 treatments.

`sid` study IDs.

`tid` treatment IDs.

`r` event counts.

`n` sample sizes.

Details

Treatment IDs represent: 1) placebo; 2) inhaled corticosteroid; 3) inhaled corticosteroid + long-acting β_2-agonist; 4) long-acting β_2-agonist; and 5) tiotropium.

Source

Baker WL, Baker EL, Coleman CI (2009). "Pharmacologic treatments for chronic obstructive pulmonary disease: a mixed-treatment comparison meta-analysis." *Pharmacotherapy*, **29**(8), 891–905. <doi:10.1592/phco.29.8.891>

dat.barlow — *A Meta-Analysis on the Effect of Parent Training Programs vs. Control for Improving Parental Psychosocial Health Within 4 Weeks After Intervention*

Description

This dataset serves as an example of meta-analysis of standardized mean differences.

Usage

```
data("dat.barlow")
```

Format

A data frame with 26 studies with the following 5 variables within each study.

`y` point esimates of standardized mean differences.

`s2` sample variances of standardized mean differences.

`n1` sample sizes in treatment group 1 (parent training programs).

`n2` sample sizes in treatment group 2 (control).

`n` total sample sizes.

Source

Barlow J, Smailagic N, Huband N, Roloff V, Bennett C (2014). "Group-based parent training programmes for improving parental psychosocial health." *Cochrane Database of Systematic Reviews*, **5**, Art. No.: CD002020. <doi:10.1002/14651858.CD002020.pub4>

dat.beck17 — *A Meta-Analysis of Prevalence of Depression or Depressive Symptoms Among Medical Students*

Description

This dataset serves as an example of meta-analysis of proportions.

Usage

```
data("dat.beck17")
```

Format

A data frame with 6 studies with the following 2 variables within each study.

`e` event counts of samples with depression or depressive symptoms.

`n` sample sizes.

Details

The original article by Rotenstein et al. (2016) stratified all extracted studies based on various screening instruments and cutoff scores. This dataset focuses on the meta-analysis of 6 studies with Beck Depression Inventory Score $\geq$17.

Source

Rotenstein LS, Ramos MA, Torre M, Segal JB, Peluso MJ, Guille C, Sen S, Mata DA (2016). "Prevalence of depression, depressive symptoms, and suicidal ideation among medical students: a systematic review and meta-analysis." *JAMA*, **316**(21), 2214–2236. <doi:10.1001/jama.2016.17324>

`dat.bellamy` *A Meta-Analysis on Type 2 Diabetes Mellitus After Gestational Diabetes*

Description

This meta-analysis serves as an example of meta-analysis with binary outcomes.

Usage

```
data("dat.bellamy")
```

Format

A data frame containing 20 cohort studies with the following 4 variables.

`sid` study IDs.

`tid` treatment/exposure IDs (0: non-exposure; 1: exposure).

`e` event counts.

`n` sample sizes.

Source

Bellamy L, Casas JP, Hingorani AD, Williams D (2009). "Type 2 diabetes mellitus after gestational diabetes: a systematic review and meta-analysis." *Lancet*, **373**(9677), 1773–1779. <doi:10.1016/S01406736(09)607315>

`dat.bjelakovic` *A Meta-Analysis on the Beneficial and Harmful Effects of Vitamin D Supplementation for the Prevention of Mortality in Healthy Adults and Adults in a Stable Phase of Disease*

Description

This meta-analysis serves as an example to illustrate the usage of the function `meta.pen`.

Usage

```
data("dat.ha")
```

Format

A data frame containing 53 studies.

`n1` the sample size in the treatment group in each study.

`n2` the sample size in the control group in each study.

`r1` the event count in the treatment group in each study.

`r2` the event count in the control group in each study.

`y` the observed effect size (log odds ratio) for each study in the meta-analysis.

`s2` the within-study variance for each study.

Source

Bjelakovic G, Gluud LL, Nikolova D, Whitfield K, Wetterslev J, Simonetti RG, Bjelakovic M, Gluud C (2014). "Vitamin D supplementation for prevention of mortality in adults." *Cochrane Database of Systematic Reviews*, **7**. Art. No.: CD007470. <doi:10.1002/14651858.CD007470.pub3>

`dat.bohren` *A Meta-Analysis on the Effects of Continuous, One-To-One Intrapartum Support on Spontaneous Vaginal Births, Compared With Usual Care*

Description

This meta-analysis serves as an example to illustrate the usage of the function `meta.pen`.

Usage

```
data("dat.ha")
```

Format

A data frame containing 21 studies.

`n1` the sample size in the treatment group in each study.

`n2` the sample size in the control group in each study.

`r1` the event count in the treatment group in each study.

`r2` the event count in the control group in each study.

`y` the observed effect size (log odds ratio) for each study in the meta-analysis.

`s2` the within-study variance for each study.

Source

Bohren MA, Hofmeyr GJ, Sakala C, Fukuzawa RK, Cuthbert A (2017). "Continuous support for women during childbirth." *Cochrane Database of Systematic Reviews*, **7**. Art. No.: CD003766. <doi:10.1002/14651858.CD003766.pub6>

---

dat.butters — *A Meta-Analysis on the Overall Response of the Addition of Drugs to a Chemotherapy Regimen for Metastatic Breast Cancer*

Description

This dataset serves as an example of meta-analysis of (log) odds ratios.

Usage

```
data("dat.butters")
```

Format

A data frame with 16 studies with the following 7 variables within each study.

`y` point esimates of log odds ratios.

`s2` sample variances of log odds ratios.

`n1` sample sizes in treatment group 1 (addition of drug).

`n2` sample sizes in treatment group 2 (control).

`r1` event counts in treatment group 1.

`r2` event counts in treatment group 2.

`n` total sample sizes.

Source

Butters DJ, Ghersi D, Wilcken N, Kirk SJ, Mallon PT (2010). "Addition of drug/s to a chemotherapy regimen for metastatic breast cancer." *Cochrane Database of Systematic Reviews*, **11**, Art. No.: CD003368. <doi:10.1002/14651858.CD003368.pub3>

dat.carless — *A Meta-Analysis on the Efficacy of Platelet-Rich-Plasmapheresis in Reducing Peri-Operative Allogeneic Red Blood Cell Transfusion in Cardiac Surgery*

Description

This meta-analysis serves as an example to illustrate the usage of the function `meta.pen`.

Usage

```
data("dat.ha")
```

Format

A data frame containing 20 studies.

`n1` the sample size in the treatment group in each study.

`n2` the sample size in the control group in each study.

`r1` the event count in the treatment group in each study.

`r2` the event count in the control group in each study.

`y` the observed effect size (log odds ratio) for each study in the meta-analysis.

`s2` the within-study variance for each study.

Source

Carless PA, Rubens FD, Anthony DM, O'Connell D, Henry DA (2011). "Platelet-rich-plasmapheresis for minimising peri-operative allogeneic blood transfusion." *Cochrane Database of Systematic Reviews*, **3**. Art. No.: CD004172. <doi:10.1002/14651858.CD004172.pub2>

dat.chor — *A Meta-Analysis of Proportions on Chorioamnionitis*

Description

This dataset serves as an example of meta-analysis of proportions.

Usage

```
data("dat.chor")
```

Format

A data frame with 21 studies with the following 2 variables within each study.

`e` event counts of horioamnionitis.

`n` sample sizes.

Source

Woodd SL, Montoya A, Barreix M, Pi L, Calvert C, Rehman AM, Chou D, Campbell OMR (2019). "Incidence of maternal peripartum infection: a systematic review and meta-analysis." *PLOS Medicine*, **16**(12), e1002984. <doi:10.1371/journal.pmed.1002984>

`dat.dep` *A Meta-Analysis of Binary and Continuous Outcomes on Depression*

Description

This dataset serves as an example of meta-analysis of combining standardized mean differences and odds ratios.

Usage

```
data("dat.dep")
```

Format

A data frame with 6 studies with the following 15 variables within each study.

`author` The first author of each study.

`year` The publication year of each study.

`treatment` The treatment group.

`control` The control group.

`y1` The sample mean in the treatment group for the continuous outcome.

`sd1` The sample standard deviation in the treatment group for the continuous outcome.

`n1` The sample size in the treatment group for the continuous outcome.

`y0` The sample mean in the control group for the continuous outcome.

`sd0` The sample standard deviation in the control group for the continuous outcome.

`n0` The sample size in the control group for the continuous outcome.

`r1` The event count in the treatment group for the binary outcome.

`m1` The sample size in the treatment group for the binary outcome.

`r0` The event count in the control group for the binary outcome.

`m0` The sample size in the control group for the binary outcome.

`id.bin` An indicator of whether the outcome is binary (1) or continuous (0).

Details

This dataset is from Cipriani et al. (2016), comparing the efficacy and tolerability of antidepressants for major depressive disorders in children and adolescents. Our case study focuses on efficacy. The authors originally performed a network meta-analysis; however, here we restrict the comparison to fluoxetine and placebo. The continuous outcomes are measured by the mean overall changes in depressive symptoms from baseline to endpoint. For the binary outcomes, events are defined as whether patients' depression rating scores were reduced by at least a specified cutoff value.

Source

Cipriani A, Zhou X, Del Giovane C, Hetrick SE, Qin B, Whittington C, Coghill D, Zhang Y, Hazell P, Leucht S, Cuijpers P, Pu J, Cohen D, Ravindran AV, Liu Y, Michael KD, Yang L, Liu L, Xie P (2016). "Comparative efficacy and tolerability of antidepressants for major depressive disorder in children and adolescents: a network meta-analysis." *The Lancet*, **388**(10047), 881–890. <doi:10.1016/S01406736(16)303853>

`dat.ducharme` *A Meta-Analysis on the Effect of Long-Acting Inhaled Beta2-Agonists vs. Control for Chronic Asthma*

Description

This meta-analysis serves as an example of meta-analysis with binary outcomes.

Usage

```
data("dat.ducharme")
```

Format

A data frame containing 33 studies with the following 4 variables within each study.

`n00` counts of non-events in treatment group 0 (placebo).

`n01` counts of events in treatment group 0 (placebo).

`n10` counts of non-events in treatment group 1 (beta2-agonists).

`n11` counts of events in treatment group 1 (beta2-agonists).

Note

The original review collected 35 studies; two double-zero-counts studies are excluded from this dataset because their odds ratios are not estimable.

Source

Ducharme FM, Ni Chroinin M, Greenstone I, Lasserson TJ (2010). "Addition of long-acting beta2-agonists to inhaled corticosteroids versus same dose inhaled corticosteroids for chronic asthma in adults and children." *Cochrane Database of Systematic Reviews*, **5**, Art. No.: CD005535. <doi:10.1002/14651858.CD005535.pub2>

dat.fib — *A Multivariate Meta-Analysis by the Fibrinogen Studies Collaboration*

Description

This multivariate meta-analysis serves as an example to illustrate function usage in the package **altmeta**. It consists of 31 studies with 4 outcomes.

Usage

```
data("dat.fib")
```

Format

A list containing three elements, `y`, `S`, and `sd`.

`y` a 31 x 4 numeric matrix containing the observed effect sizes; the rows represent studies and the columns represent outcomes.

`S` a list containing 31 elements; each element is within-study covariance matrix of the corresponding study.

`sd` a 31 x 4 numeric matrix containing the within-study standard deviations; the rows represent studies and the columns represent outcomes.

Source

Fibrinogen Studies Collaboration (2004). "Collaborative meta-analysis of prospective studies of plasma fibrinogen and cardiovascular disease." *European Journal of Cardiovascular Prevention and Rehabilitation*, **11**(1), 9–17. <doi:10.1097/01.hjr.0000114968.39211.01>

Fibrinogen Studies Collaboration (2005). "Plasma fibrinogen level and the risk of major cardiovascular diseases and nonvascular mortality: an individual participant meta-analysis." *JAMA*, **294**(14), 1799–1809. <doi:10.1001/jama.294.14.1799>

dat.ha — *A Meta-Analysis on the Effect of Placebo Interventions for All Clinical Conditions Regarding Patient-Reported Outcomes*

Description

This meta-analysis serves as an example to illustrate function usage in the package **altmeta**.

Usage

```
data("dat.ha")
```

Format

A data frame containing 109 studies with the observed effect sizes and their within-study variances.

`y` the observed effect size for each collected study in the meta-analysis.

`s2` the within-study variance for each study.

Source

Hrobjartsson A, Gotzsche PC (2010). "Placebo interventions for all clinical conditions." *Cochrane Database of Systematic Reviews*, **1**. Art. No.: CD003974. <doi:10.1002/14651858.CD003974.pub3>

`dat.henry` *A Meta-Analysis for Evaluating the Effect of Tranexamic Acid on Perioperative Allogeneic Blood Transfusion*

Description

This meta-analysis serves as an example of meta-analysis with binary outcomes.

Usage

```
data("dat.henry")
```

Format

A data frame containing 26 studies with the following 4 variables within each study.

`n00` counts of non-events in treatment group 0 (placebo).

`n01` counts of events in treatment group 0 (placebo).

`n10` counts of non-events in treatment group 1 (tranexamic acid).

`n11` counts of events in treatment group 1 (tranexamic acid).

Note

The original review collected 27 studies; one double-zero-counts study is excluded from this dataset because its odds ratio is not estimable.

Source

Henry DA, Carless PA, Moxey AJ, O'Connell, Stokes BJ, Fergusson DA, Ker K (2011). "Anti-fibrinolytic use for minimising perioperative allogeneic blood transfusion." *Cochrane Database of Systematic Reviews*, **1**, Art. No.: CD001886. <doi:10.1002/14651858.CD001886.pub3>

dat.hipfrac — *A Meta-Analysis on the Magnitude and Duration of Excess Mortality After Hip Fracture Among Older Men*

Description

This meta-analysis serves as an example to illustrate function usage in the package **altmeta**.

Usage

```
data("dat.hipfrac")
```

Format

A data frame containing 17 studies with the observed effect sizes and their within-study variances.

`y` the observed effect size for each collected study in the meta-analysis.

`s2` the within-study variance for each study.

Source

Haentjens P, Magaziner J, Colon-Emeric CS, Vanderschueren D, Milisen K, Velkeniers B, Boonen S (2010). "Meta-analysis: excess mortality after hip fracture among older women and men". *Annals of Internal Medicine*, **152**(6), 380–390. <doi:10.7326/0003481915262010031600008>

dat.hughes — *A Meta-Analysis on the Effects of Ovulation Suppression for Endometriosis in Subfertile Women*

Description

This meta-analysis serves as an example to illustrate the usage of the function `metahet.hybrid`.

Usage

```
data("dat.hughes")
```

Format

A data frame containing 11 studies.

`y` the observed effect size (log odds ratio) for each study in the meta-analysis.

`s2` the within-study variance for each study.

Source

Hughes E, Brown J, Collins JJ, et al. (2007). "Ovulation suppression for endometriosis for women with subfertility." *Cochrane Database of Systematic Reviews*, **7**. Art. No.: CD000155. <doi:10.1002/14651858.CD000155.pub2>

`dat.kaner` *A Meta-Analysis on the Effect of Brief Alcohol Interventions vs. Control in Primary Care Populations*

Description

This dataset serves as an example of meta-analysis of risk differences.

Usage

```
data("dat.kaner")
```

Format

A data frame with 13 studies with the following 7 variables within each study.

`y` point esimates of risk differences.

`s2` sample variances of risk differences.

`n1` sample sizes in treatment group 1 (brief alcohol interventions).

`n2` sample sizes in treatment group 2 (control).

`r1` event counts in treatment group 1.

`r2` event counts in treatment group 2.

`n` total sample sizes.

Source

Kaner EF, Dickinson HO, Beyer FR, Campbell F, Schlesinger C, Heather N, Saunders JB, Burnand B, Pienaar ED (2007). "Effectiveness of brief alcohol interventions in primary care populations." *Cochrane Database of Systematic Reviews*, **2**, Art. No.: CD004148. <doi:10.1002/14651858.CD004148.pub3>

dat.lcj — *A Meta-Analysis on the Effect of Progressive Resistance Strength Training Exercise vs. Control*

Description

This meta-analysis serves as an example to illustrate function usage in the package **altmeta**.

Usage

```
data("dat.lcj")
```

Format

A data frame containing 33 studies with the observed effect sizes and their within-study variances.

`y` the observed effect size for each collected study in the meta-analysis.

`s2` the within-study variance for each study.

Source

Liu CJ, Latham NK (2009). "Progressive resistance strength training for improving physical function in older adults." *Cochrane Database of Systematic Reviews*, **3**. Art. No.: CD002759. <doi:10.1002/14651858.CD002759.pub2>

dat.paige — *A Meta-Analysis on the Effectiveness of Spinal Manipulative Therapies (Other Than Sham)*

Description

This dataset serves as an example of meta-analysis of standardized mean differences.

Usage

```
data("dat.paige")
```

Format

A data frame with 6 studies with the following 5 variables within each study.

`y` point esimates of standardized mean differences.

`s2` sample variances of standardized mean differences.

`n1` sample sizes in treatment group 1 (spinal manipulation).

`n2` sample sizes in treatment group 2 (comparator).

`n` total sample sizes.

Source

Paige NM, Miake-Lye IM, Booth MS, Beroes JM, Mardian AS, Dougherty P, Branson R, Tang B, Morton SC, Shekelle PG (2017). "Association of spinal manipulative therapy with clinical benefit and harm for acute low back pain: systematic review and meta-analysis." *JAMA*, **317**(14), 1451–1460. <doi:10.1001/jama.2017.3086>

`dat.plourde` *A Meta-Analysis for Comparing the Fluoroscopy Time in Percutaneous Coronary Intervention Between Radial and Femoral Accesses*

Description

This dataset serves as an example of meta-analysis of mean differences.

Usage

```
data("dat.plourde")
```

Format

A data frame with 19 studies with the following 5 variables within each study.

`y` point esimates of mean differences.

`s2` sample variances of mean differences.

`n1` sample sizes in treatment group 1 (radial).

`n2` sample sizes in treatment group 2 (femoral).

`n` total sample sizes.

Source

Plourde G, Pancholy SB, Nolan J, Jolly S, Rao SV, Amhed I, Bangalore S, Patel T, Dahm JB, Bertrand OF (2015). "Radiation exposure in relation to the arterial access site used for diagnostic coronary angiography and percutaneous coronary intervention: a systematic review and meta-analysis." *Lancet*, **386**(10009), 2192–2203. <doi:10.1016/S01406736(15)003050>

dat.poole — *A Meta-Analysis for Evaluating the Effect of Mucolytic on Bronchitis/Chronic Obstructive Pulmonary Disease*

Description

This meta-analysis serves as an example of meta-analysis with binary outcomes.

Usage

```
data("dat.poole")
```

Format

A data frame containing 24 studies with the following 4 variables within each study.

`n00` counts of non-events in treatment group 0 (placebo).

`n01` counts of events in treatment group 0 (placebo).

`n10` counts of non-events in treatment group 1 (mucolytic).

`n11` counts of events in treatment group 1 (mucolytic).

Source

Poole P, Chong J, Cates CJ (2015). "Mucolytic agents versus placebo for chronic bronchitis or chronic obstructive pulmonary disease." *Cochrane Database of Systematic Reviews*, **7**, Art. No.: CD001287. <doi:10.1002/14651858.CD001287.pub5>

dat.pte — *Meta-Analysis of Multiple Risk Factors for Pterygium*

Description

This dataset serves as an example to illustrate network meta-analysis of multiple factors. It consists of 29 studies on a total of 8 risk factors: area of residence (rural vs. urban); education attainment (low vs. high); latitude of residence (low vs. high); occupation type (outdoor vs. indoor); smoking status (yes vs. no); use of hat (yes vs. no); use of spectacles (yes vs. no); and use of sunglasses (yes vs. no). Each study only investigates a subset of the 8 risk factors, so the dataset contains many missing values.

Usage

```
data("dat.pte")
```

Format

A list containing two elements, `y` and `se`.

`y` a 29 x 8 numeric matrix containing the observed effect sizes; the rows represent studies and the columns represent outcomes.

`se` a 29 x 8 numeric matrix containing the within-study standard errors; the rows represent studies and the columns represent outcomes.

Source

Serghiou S, Patel CJ, Tan YY, Koay P, Ioannidis JPA (2016). "Field-wide meta-analyses of observational associations can map selective availability of risk factors and the impact of model specifications." *Journal of Clinical Epidemiology*, **71**, 58–67. <doi:10.1016/j.jclinepi.2015.09.004>

---

dat.sc *A Network Meta-Analysis on Smoking Cessation*

Description

The dataset is extracted from Lu and Ades (2006); it was initially reported by Hasselblad (1998) (without performing a formal network meta-analysis).

Usage

```
data("dat.sc")
```

Format

A dataset of network meta-analysis with binary outcomes, containing 24 studies and 4 treatments.

`sid` study IDs.

`tid` treatment IDs.

`r` event counts.

`n` sample sizes.

Details

Treatment IDs represent: 1)no contact; 2) selfhelp; 3) individual counseling; and 4) group counseling.

Source

Hasselblad V (1998). "Meta-analysis of multitreatment studies." *Medical Decision Making*, **18**(1), 37–43. <doi:10.1177/0272989X9801800110>

Lu G, Ades AE (2006). "Assessing evidence inconsistency in mixed treatment comparisons." *Journal of the American Statistical Association*, **101**(474), 447–459. <doi:10.1198/016214505000001302>

dat.scheidler — *Meta-Analysis on the Utility of Lymphangiography, Computed Tomography, and Magnetic Resonance Imaging for the Diagnosis of Lymph Node Metastasis*

Description

This meta-analysis serves as an example of meta-analyses of diagnostic tests.

Usage

```
data("dat.scheidler")
```

Format

A data frame with 44 studies with the following 5 variables; each row represents a study.

`dt` types of diagnostic tests; CT: computed tomography; LAG: lymphangiography; and MRI: magnetic resonance imaging.

`tp` counts of true positives.

`fp` counts of false positives.

`fn` counts of false negatives.

`tn` counts of true negatives.

Source

Scheidler J, Hricak H, Yu KK, Subak L, Segal MR (1997). "Radiological evaluation of lymph node metastases in patients with cervical cancer: a meta-analysis." *JAMA*, **278**(13), 1096–1101. <doi:10.1001/jama.1997.03550130070040>

dat.slf — *A Meta-Analysis on the Effect of Nicotine Gum for Smoking Cessation*

Description

This meta-analysis serves as an example to illustrate function usage in the package **altmeta**.

Usage

```
data("dat.slf")
```

Format

A data frame containing 56 studies with the observed effect sizes and their within-study variances.

`y` the observed effect size for each collected study in the meta-analysis.

`s2` the within-study variance for each study.

Source

Stead LF, Perera R, Bullen C, Mant D, Hartmann-Boyce J, Cahill K, Lancaster T (2012). "Nicotine replacement therapy for smoking cessation." *Cochrane Database of Systematic Reviews*, **11**. Art. No.: CD000146. <doi:10.1002/14651858.CD000146.pub4>

`dat.smith` *Meta-Analysis on the Diagnostic Accuracy of Ultrasound for Detecting Partial Thickness Rotator Cuff Tears*

Description

This meta-analysis serves as an example of meta-analyses of diagnostic tests.

Usage

```
data("dat.smith")
```

Format

A data frame with 30 studies with the following 4 variables; each row represents a study.

`tp` counts of true positives.

`fp` counts of false positives.

`fn` counts of false negatives.

`tn` counts of true negatives.

Source

Smith TO, Back T, Toms AP, Hing CB (2011). "Diagnostic accuracy of ultrasound for rotator cuff tears in adults: a systematic review and meta-analysis." *Clinical Radiology*, **66**(11), 1036–1048. <doi:10.1016/j.crad.2011.05.007>

`dat.whiting` *A Meta-Analysis on Adverse Events for the Comparison Cannabinoid vs. Placebo*

Description

This dataset serves as an example of meta-analysis of (log) odds ratios.

Usage

```
data("dat.whiting")
```

Format

A data frame with 29 studies with the following 9 variables within each study.

`y` point esimates of log odds ratios.

`s2` sample variances of log odds ratios.

`n00` counts of non-events in treatment group 0 (placebo).

`n01` counts of events in treatment group 0.

`n10` counts of non-events in treatment group 1 (cannabinoid).

`n11` counts of events in treatment group 1 (cannabinoid).

`n0` sample sizes in treatment group 0.

`n1` sample sizes in treatment group 1.

`n` total sample sizes.

Source

Whiting PF, Wolff RF, Deshpande S, Di Nisio M, Duffy S, Hernandez AV, Keurentjes JC, Lang S, Misso K, Ryder S, Schmidlkofer S, Westwood M, Kleijnen J (2015). "Cannabinoids for medical use: a systematic review and meta-analysis." *JAMA*, **313**(24), 2456–2473. <doi:10.1001/jama.2015.6358>

`dat.williams` *A Meta-Analysis on the Effect of Pharmacotherapy for Social Anxiety Disorder*

Description

This dataset serves as an example of meta-analysis of (log) relative risks.

Usage

```
data("dat.williams")
```

Format

A data frame with 20 studies with the following 7 variables within each study.

`y` point esimates of log relative risks.

`s2` sample variances of log relative risks.

`n1` sample sizes in treatment group 1 (medication).

`n2` sample sizes in treatment group 2 (placebo).

`r1` event counts in treatment group 1.

`r2` event counts in treatment group 2.

`n` total sample sizes.

Source

Williams T, Hattingh CJ, Kariuki CM, Tromp SA, van Balkom AJ, Ipser JC, Stein DJ (2017). "Pharmacotherapy for social anxiety disorder (SAnD)." *Cochrane Database of Systematic Reviews*, **10**, Art. No.: CD001206. <doi:10.1002/14651858.CD001206.pub3>

`dat.xu` *A Network Meta-Analysis on Immune Checkpoint Inhibitor Drugs*

Description

This network meta-analysis investigates the effects of seven immune checkpoint inhibitor (ICI) drugs on all-grade treatment-related adverse events (TrAEs). It aimed to provide a safety ranking of the ICI drugs for the treatment of cancer.

Usage

```
data("dat.xu")
```

Format

A dataset of network meta-analysis with binary outcomes, containing 23 studies and 7 treatments.

`sid` study IDs.

`tid` treatment IDs.

`r` event counts.

`n` sample sizes.

Details

Treatment IDs represent: 1) conventional therapy; 2) nivolumab; 3) pembrolizumab; 4) two ICIs; 5) ICI and conventional therapy; 6) atezolizumab; and 7) ipilimumab.

Source

Xu C, Chen YP, Du XJ, Liu JQ, Huang CL, Chen L, Zhou GQ, Li WF, Mao YP, Hsu C, Liu Q, Lin AH, Tang LL, Sun Y, Ma J (2018). "Comparative safety of immune checkpoint inhibitors in cancer: systematic review and network meta-analysis." *BMJ*, **363**, k4226. <doi:10.1136/bmj.k4226>

maprop.glmm *Meta-Analysis of Proportions Using Generalized Linear Mixed Models*

Description

Performs a meta-analysis of proportions using generalized linear mixed models (GLMMs) with various link functions.

Usage

```
maprop.glmm(e, n, data, link = "logit", alpha = 0.05,
            pop.avg = TRUE, int.approx = 10000, b.iter = 1000,
            seed = 1234, ...)
```

Arguments

e	a numeric vector specifying the event counts in the collected studies.
n	a numeric vector specifying the sample sizes in the collected studies.
data	an optional data frame containing the meta-analysis dataset. If data is specified, the previous arguments, e and n, should be specified as their corresponding column names in data.
link	a character string specifying the link function used in the GLMM, which can be one of "log" (log link), "logit" (logit link, the default), "probit" (probit link), "cauchit" (cauchit link), and "cloglog" (complementary log-log link).
alpha	a numeric value specifying the statistical significance level.
pop.avg	a logical value indicating whether the population-averaged proportion and its confidence interval are to be produced. This quantity is the marginal mean of study-specific proportions, while the commonly-reported overall proportion usually represents the median (or interpreted as a conditional measure); see more details about this quantity in Section 13.2.3 in Agresti (2013), Chu et al. (2012), Lin and Chu (2020), and Zeger et al. (1988). If pop.avg = TRUE (the default), the bootstrap resampling is used to produce the confidence interval of the population-averaged proportion; the confidence interval of the commonly-reported median proportion will be also produced, in addition to its conventional confidence interval (by back-transforming the Wald-type confidence interval derived on the scale specified by link).
int.approx	an integer specifying the number of independent standard normal samples for numerically approximating the integration involved in the calculation of the population-averaged proportion; see details in Lin and Chu (2020). It is only used when pop.avg = TRUE and link is not "probit". The probit link leads to a closed form of the population-averaged proportion, so it does not need the numerical approximation; for other links, the population-averaged proportion does not have a closed form.

`b.iter`	an integer specifying the number of bootstrap iterations; it is only used when `pop.avg = TRUE`.
`seed`	an integer for specifying the seed of the random number generation for reproducibility during the bootstrap resampling (and numerical approximation for the population-averaged proportion); it is only used when `pop.avg = TRUE`.
`...`	other arguments that can be passed to the function `glmer` in the package **lme4**.

Value

This function returns a list containing the point and interval estimates of the overall proportion. Specifically, `prop.c.est` is the commonly-reported median (or conditional) proportion, and `prop.c.ci` is its confidence interval. It also returns information about AIC, BIC, log likelihood, deviance, and residual degrees-of-freedom. If `pop.avg = TRUE`, the following additional elements will be also in the produced list: `prop.c.ci.b` is the bootstrap confidence interval of the commonly-reported median (conditional) proportion, `prop.m.est` is the point estimate of the population-averaged (marginal) proportion, `prop.m.ci.b` is the bootstrap confidence interval of the population-averaged (marginal) proportion, and `b.w.e` is a vector of two numeric values, indicating the counts of warnings and errors occurred during the bootstrap iterations.

Note

This function implements the GLMM for the meta-analysis of proportions via the function `glmer` in the package **lme4**. It is possible that the algorithm of the GLMM estimation may not converge for some bootstrapped meta-analyses when `pop.avg = TRUE`, and the function `glmer` may report warnings or errors about the convergence issue. The bootstrap iterations are continued until `b.iter` replicates without any warnings or errors are obtained; those replicates with any warnings or errors are discarded.

References

Agresti A (2013). *Categorical Data Analysis*. Third edition. John Wiley & Sons, Hoboken, NJ.

Bakbergenuly I, Kulinskaya E (2018). "Meta-analysis of binary outcomes via generalized linear mixed models: a simulation study." *BMC Medical Research Methodology*, **18**, 70. <doi:10.1186/s1287401805319>

Chu H, Nie L, Chen Y, Huang Y, Sun W (2012). "Bivariate random effects models for meta-analysis of comparative studies with binary outcomes: methods for the absolute risk difference and relative risk." *Statistical Methods in Medical Research*, **21**(6), 621–633. <doi:10.1177/0962280210393712>

Hamza TH, van Houwelingen HC, Stijnen T (2008). "The binomial distribution of meta-analysis was preferred to model within-study variability." *Journal of Clinical Epidemiology*, **61**(1), 41–51. <doi:10.1016/j.jclinepi.2007.03.016>

Lin L, Chu H (2020). "Meta-analysis of proportions using generalized linear mixed models." *Epidemiology*, **31**(5), 713–717. <doi:10.1097/ede.0000000000001232>

Stijnen T, Hamza TH, Ozdemir P (2010). "Random effects meta-analysis of event outcome in the framework of the generalized linear mixed model with applications in sparse data." *Statistics in Medicine*, **29**(29), 3046–3067. <doi:10.1002/sim.4040>

Zeger SL, Liang K-Y, Albert PS (1988). "Models for longitudinal data: a generalized estimating equation approach." *Biometrics*, **44**(4), 1049–1060. <doi:10.2307/2531734>

See Also

maprop.twostep

Examples

```
# chorioamnionitis data
data("dat.chor")
# GLMM with the logit link with only 10 bootstrap iterations
out.chor.glmm.logit <- maprop.glmm(e, n, data = dat.chor,
  link = "logit", b.iter = 10, seed = 1234)
out.chor.glmm.logit
# not calculating the population-averaged (marginal) proportion,
#  without bootstrap resampling
out.chor.glmm.logit <- maprop.glmm(e, n, data = dat.chor,
  link = "logit", pop.avg = FALSE)
out.chor.glmm.logit

# increases the number of bootstrap iterations to 1000,
#  taking longer time
out.chor.glmm.logit <- maprop.glmm(e, n, data = dat.chor,
  link = "logit", b.iter = 1000, seed = 1234)
out.chor.glmm.logit

# GLMM with the log link
out.chor.glmm.log <- maprop.glmm(e, n, data = dat.chor,
  link = "log", b.iter = 10, seed = 1234)
out.chor.glmm.log
# GLMM with the probit link
out.chor.glmm.probit <- maprop.glmm(e, n, data = dat.chor,
  link = "probit", b.iter = 10, seed = 1234)
out.chor.glmm.probit
# GLMM with the cauchit link
out.chor.glmm.cauchit <- maprop.glmm(e, n, data = dat.chor,
  link = "cauchit", b.iter = 10, seed = 1234)
out.chor.glmm.cauchit
# GLMM with the cloglog link
out.chor.glmm.cloglog <- maprop.glmm(e, n, data = dat.chor,
  link = "cloglog", b.iter = 10, seed = 1234)
out.chor.glmm.cloglog


# depression data
data("dat.beck17")
out.beck17.glmm.log <- maprop.glmm(e, n, data = dat.beck17,
  link = "log", b.iter = 10, seed = 1234)
out.beck17.glmm.log
out.beck17.glmm.logit <- maprop.glmm(e, n, data = dat.beck17,
  link = "logit", b.iter = 10, seed = 1234)
out.beck17.glmm.logit
out.beck17.glmm.probit <- maprop.glmm(e, n, data = dat.beck17,
  link = "probit", b.iter = 10, seed = 1234)
out.beck17.glmm.probit
```

```
out.beck17.glmm.cauchit <- maprop.glmm(e, n, data = dat.beck17,
  link = "cauchit", b.iter = 10, seed = 1234)
out.beck17.glmm.cauchit
out.beck17.glmm.cloglog<- maprop.glmm(e, n, data = dat.beck17,
  link = "cloglog", b.iter = 10, seed = 1234)
out.beck17.glmm.cloglog
```

`maprop.twostep` *Meta-Analysis of Proportions Using Two-Step Methods*

Description

Performs a meta-analysis of proportions using conventional two-step methods with various data transformations.

Usage

```
maprop.twostep(e, n, data, link = "logit", method = "ML", alpha = 0.05,
               pop.avg = TRUE, int.approx = 10000, b.iter = 1000,
               seed = 1234)
```

Arguments

`e`	a numeric vector specifying the event counts in the collected studies.
`n`	a numeric vector specifying the sample sizes in the collected studies.
`data`	an optional data frame containing the meta-analysis dataset. If `data` is specified, the previous arguments, `e` and `n`, should be specified as their corresponding column names in `data`.
`link`	a character string specifying the data transformation for each study's proportion used in the two-step method, which can be one of `"log"` (log transformation), `"logit"` (logit transformation, the default), `"arcsine"` (arcsine transformation), and `"double.arcsine"` (Freeman–Tukey double-arcsine transformation).
`method`	a character string specifying the method to perform the meta-analysis, which is passed to the argument `method` in the function `rma.uni` in the package **metafor**. It can be one of `"ML"` (maximum likelihood, the default), `"REML"` (restricted maximum likelihood), and many other options; see more details in the manual of **metafor**. The default is set to `"ML"` for consistency with the function `maprop.glmm`, where generalized linear mixed models are often estimated via the maximum likelihood approach. For the two-step method, users might also use `"REML"` because the restricted maximum likelihood estimation may have superior performance in many cases.
`alpha`	a numeric value specifying the statistical significance level.

`pop.avg` a logical value indicating whether the population-averaged proportion and its confidence interval are to be produced. This quantity is the marginal mean of study-specific proportions, while the commonly-reported overall proportion usually represents the median (or interpreted as a conditional measure); see more details about this quantity in Section 13.2.3 in Agresti (2013), Chu et al. (2012), Lin and Chu (2020), and Zeger et al. (1988). If `pop.avg = TRUE` (the default), the bootstrap resampling is used to produce the confidence interval of the population-averaged proportion; the confidence interval of the commonly-reported median proportion will be also produced, in addition to its conventional confidence interval (by back-transforming the Wald-type confidence interval derived on the scale specified by `link`).

`int.approx` an integer specifying the number of independent standard normal samples for numerically approximating the integration involved in the calculation of the population-averaged proportion; see details in Lin and Chu (2020). It is only used when `pop.avg = TRUE`. For the commonly-used data transformations available for `link`, the population-averaged proportion does not have a closed form.

`b.iter` an integer specifying the number of bootstrap iterations; it is only used when `pop.avg = TRUE`.

`seed` an integer for specifying the seed of the random number generation for reproducibility during the bootstrap resampling (and numerical approximation for the population-averaged proportion); it is only used when `pop.avg = TRUE`.

Value

This function returns a list containing the point and interval estimates of the overall proportion. Specifically, `prop.c.est` is the commonly-reported median (or conditional) proportion, and `prop.c.ci` is its confidence interval. If `pop.avg = TRUE`, the following additional elements will be also in the produced list: `prop.c.ci.b` is the bootstrap confidence interval of the commonly-reported median (conditional) proportion, `prop.m.est` is the point estimate of the population-averaged (marginal) proportion, `prop.m.ci.b` is the bootstrap confidence interval of the population-averaged (marginal) proportion, and `b.w.e` is a vector of two numeric values, indicating the counts of warnings and errors occurred during the bootstrap iterations. Moreover, if the Freeman–Tukey double-arcsine transformation (`link = "double.arcsine"`) is used, the back-transformation will be implemented at four values as the overall sample size: the harmonic, geometric, and arithmetic means of the study-specific sample sizes, and the inverse of the synthesized result's variance. See details in Barendregt et al. (2013) and Schwarzer et al. (2019).

Note

This function implements the two-step method for the meta-analysis of proportions via the `rma.uni` function in the package **metafor**. It is possible that the algorithm of the maximum likelihood or restricted maximum likelihood estimation may not converge for some bootstrapped meta-analyses when `pop.avg = TRUE`, and the `rma.uni` function may report warnings or errors about the convergence issue. The bootstrap iterations are continued until `b.iter` replicates without any warnings or errors are obtained; those replicates with any warnings or errors are discarded.

References

Agresti A (2013). *Categorical Data Analysis*. Third edition. John Wiley & Sons, Hoboken, NJ.

Barendregt JJ, Doi SA, Lee YY, Norman RE, Vos T (2013). "Meta-analysis of prevalence." *Journal of Epidemiology and Community Health*, **67**(11), 974–978. <doi:10.1136/jech2013203104>

Freeman MF, Tukey JW (1950). "Transformations related to the angular and the square root." *The Annals of Mathematical Statistics*, **21**(4), 607–611. <doi:10.1214/aoms/1177729756>

Lin L, Chu H (2020). "Meta-analysis of proportions using generalized linear mixed models." *Epidemiology*, **31**(5), 713–717. <doi:10.1097/ede.0000000000001232>

Miller JJ (1978). "The inverse of the Freeman–Tukey double arcsine transformation." *The American Statistician*, **32**(4), 138. <doi:10.1080/00031305.1978.10479283>

Schwarzer G, Chemaitelly H, Abu-Raddad LJ, Rucker G (2019). "Seriously misleading results using inverse of Freeman-Tukey double arcsine transformation in meta-analysis of single proportions." *Research Synthesis Methods*, **10**(3), 476–483. <doi:10.1002/jrsm.1348>

Viechtbauer W (2010). "Conducting meta-analyses in R with the metafor package." *Journal of Statistical Software*, **36**, 3. <doi:10.18637/jss.v036.i03>

Zeger SL, Liang K-Y, Albert PS (1988). "Models for longitudinal data: a generalized estimating equation approach." *Biometrics*, **44**(4), 1049–1060. <doi:10.2307/2531734>

See Also

maprop.glmm

Examples

```
# chorioamnionitis data
data("dat.chor")
# two-step method with the logit transformation
out.chor.twostep.logit <- maprop.twostep(e, n, data = dat.chor,
  link = "logit", b.iter = 10, seed = 1234)
out.chor.twostep.logit
# not calculating the population-averaged (marginal) proportion,
#  without bootstrap resampling
out.chor.twostep.logit <- maprop.twostep(e, n, data = dat.chor,
  link = "logit", pop.avg = FALSE)
out.chor.twostep.logit

# increases the number of bootstrap iterations to 1000,
#  taking longer time
out.chor.twostep.logit <- maprop.twostep(e, n, data = dat.chor,
  link = "logit", b.iter = 1000, seed = 1234)
out.chor.twostep.logit

# two-step method with the log transformation
out.chor.twostep.log <- maprop.twostep(e, n, data = dat.chor,
  link = "log", b.iter = 10, seed = 1234)
out.chor.twostep.log
# two-step method with the arcsine transformation
out.chor.twostep.arcsine <- maprop.twostep(e, n, data = dat.chor,
  link = "arcsine", b.iter = 10, seed = 1234)
out.chor.twostep.arcsine
# two-step method with the Freeman--Tukey double-arcsine transformation
```

```
out.chor.twostep.double.arcsine <- maprop.twostep(e, n, data = dat.chor,
  link = "double.arcsine", b.iter = 10, seed = 1234)
out.chor.twostep.double.arcsine


# depression data
data("dat.beck17")
out.beck17.twostep.log <- maprop.twostep(e, n, data = dat.beck17,
  link = "log", b.iter = 10, seed = 1234)
out.beck17.twostep.log
out.beck17.twostep.logit <- maprop.twostep(e, n, data = dat.beck17,
  link = "logit", b.iter = 10, seed = 1234)
out.beck17.twostep.logit
out.beck17.twostep.arcsine <- maprop.twostep(e, n, data = dat.beck17,
  link = "arcsine", b.iter = 10, seed = 1234)
out.beck17.twostep.arcsine
out.beck17.twostep.double.arcsine <- maprop.twostep(e, n, data = dat.beck17,
  link = "double.arcsine", b.iter = 10, seed = 1234)
out.beck17.twostep.double.arcsine
```

`meta.biv` *Bivariate Method for Meta-Analysis.*

Description

Performs a meta-analysis with a binary outcome using a bivariate generalized linear mixed model (GLMM) described in Chu et al. (2012).

Usage

```
meta.biv(sid, tid, e, n, data, link = "logit", alpha = 0.05,
         b.iter = 1000, seed = 1234, ...)
```

Arguments

`sid`	a vector specifying the study IDs.
`tid`	a vector of 0/1 specifying the treatment/exposure IDs (0: control/non-exposure; 1: treatment/exposure).
`e`	a numeric vector specifying the event counts.
`n`	a numeric vector specifying the sample sizes.
`data`	an optional data frame containing the meta-analysis dataset. If `data` is specified, the previous arguments, `sid`, `tid`, `e`, and `n`, should be specified as their corresponding column names in `data`.
`link`	a character string specifying the link function used in the GLMM, which can be either `"logit"` (the default) or `"probit"`.
`alpha`	a numeric value specifying the statistical significance level.

b.iter	an integer specifying the number of bootstrap iterations, which are used to produce confidence intervals of marginal results.
seed	an integer for specifying the seed of the random number generation for reproducibility during the bootstrap resampling.
...	other arguments that can be passed to the function glmer in the package **lme4**.

Details

Suppose a meta-analysis with a binary outcome contains N studies. Let n_{i0} and n_{i1} be the sample sizes in the control/non-exposure and treatment/exposure groups in study i, respectively, and let e_{i0} and e_{i1} be the event counts ($i = 1, \ldots, N$). The event counts are assumed to independently follow binomial distributions:

$$e_{i0} \sim Bin(n_{i0}, p_{i0});$$

$$e_{i1} \sim Bin(n_{i1}, p_{i1}),$$

where p_{i0} and p_{i1} represent the true event probabilities. They are modeled jointly as follows:

$$g(p_{i0}) = \mu_0 + \nu_{i0};$$

$$g(p_{i1}) = \mu_1 + \nu_{i1};$$

$$(\nu_{i0}, \nu_{i1})' \sim N((0,0)', \mathbf{\Sigma}).$$

Here, $g(\cdot)$ denotes the link function that transforms the event probabilities to linear forms. The fixed effects μ_0 and μ_1 represent the overall event probabilities on the transformed scale. The study-specific parameters ν_{i0} and ν_{i1} are random effects, which are assumed to follow the bivariate normal distribution with zero means and variance-covariance matrix $\mathbf{\Sigma}$. The diagonal elements of $\mathbf{\Sigma}$ are σ_0^2 and σ_1^2 (between-study variances due to heterogeneity), and the off-diagonal elements are $\rho\sigma_0\sigma_1$, where ρ is the correlation coefficient.

When using the logit link, $\mu_1 - \mu_0$ represents the log odds ratio (Van Houwelingen et al., 1993; Stijnen et al., 2010; Jackson et al., 2018); $\exp(\mu_1 - \mu_0)$ may be referred to as the conditional odds ratio (Agresti, 2013). Alternatively, we can obtain the marginal event probabilities (Chu et al., 2012):

$$p_k = E[p_{ik}] \approx \left[1 + \exp\left(-\mu_k / \sqrt{1 + C^2\sigma_k^2}\right)\right]^{-1}$$

for k = 0 and 1, where $C = 16\sqrt{3}/(15\pi)$. The marginal odds ratio, relative risk, and risk difference are subsequently obtained as $[p_1/(1-p_1)]/[p_0/(1-p_0)]$, p_1/p_0, and $p_1 - p_0$, respectively.

When using the probit link, the model does not yield the conditional odds ratio. The marginal probabilities have closed-form solutions:

$$p_k = E[p_{ik}] = \Phi\left(\mu_k / \sqrt{1 + \sigma_k^2}\right)$$

for k = 0 and 1, where $\Phi(\cdot)$ is the cumulative distribution function of the standard normal distribution. They further lead to the marginal odds ratio, relative risk, and risk difference.

Value

This function returns a list containing the point and interval estimates of the marginal event rates (`p0.m`, `p0.m.ci`, `p1.m`, and `p1.m.ci`), odds ratio (`OR.m` and `OR.m.ci`), relative risk (`RR.m` and `RR.m.ci`), risk difference (`RD.m` and `RD.m.ci`), and correlation coefficient between the two treatment/exposure groups (`rho` and `rho.ci`). These interval estimates are obtained using the bootstrap resampling. During the bootstrap resampling, computational warnings or errors may occur for implementing the bivariate GLMM in some resampled meta-analyses. This function returns the counts of warnings and errors (`b.w.e`). The resampled meta-analyses that lead to warnings and errors are not used for producing the bootstrap confidence intervals; the bootstrap iterations stop after obtaining `b.iter` resampled meta-analyses without warnings and errors. If the logit link is used (`link = "logit"`), it also returns the point and interval estimates of the conditional odds ratio (`OR.c` and `OR.c.ci`), which are more frequently reported in the current literature than the marginal odds ratios. Unlike the marginal results that use the bootstrap resampling to produce their confidence intervals, the Wald-type confidence interval is calculated for the log conditional odds ratio; it is then transformed to the odds ratio scale.

References

Agresti A (2013). *Categorical Data Analysis*. Third edition. John Wiley & Sons, Hoboken, NJ.

Chu H, Nie L, Chen Y, Huang Y, Sun W (2012). "Bivariate random effects models for meta-analysis of comparative studies with binary outcomes: methods for the absolute risk difference and relative risk." *Statistical Methods in Medical Research*, **21**(6), 621–633. <doi:10.1177/0962280210393712>

Jackson D, Law M, Stijnen T, Viechtbauer W, White IR (2018). "A comparison of seven random-effects models for meta-analyses that estimate the summary odds ratio." *Statistics in Medicine*, **37**(7), 1059–1085. <doi:10.1002/sim.7588>

Stijnen T, Hamza TH, Ozdemir P (2010). "Random effects meta-analysis of event outcome in the framework of the generalized linear mixed model with applications in sparse data." *Statistics in Medicine*, **29**(29), 3046–3067. <doi:10.1002/sim.4040>

Van Houwelingen HC, Zwinderman KH, Stijnen T (1993). "A bivariate approach to meta-analysis." *Statistics in Medicine*, **12**(24), 2273–2284. <doi:10.1002/sim.4780122405>

See Also

`maprop.glmm`, `meta.dt`

Examples

```
data("dat.bellamy")
out.bellamy.logit <- meta.biv(sid, tid, e, n, data = dat.bellamy,
  link = "logit", b.iter = 1000)
out.bellamy.logit
out.bellamy.probit <- meta.biv(sid, tid, e, n, data = dat.bellamy,
  link = "probit", b.iter = 1000)
out.bellamy.probit
```

`meta.dt` *Meta-Analysis of Diagnostic Tests*

Description

Performs a meta-analysis of diagnostic tests using approaches described in Reitsma et al. (2005) and Chu and Cole (2006).

Usage

```
meta.dt(tp, fp, fn, tn, data, method = "biv.glmm", alpha = 0.05, ...)
```

Arguments

`tp`	counts of true positives.
`fp`	counts of false positives.
`fn`	counts of false negatives.
`tn`	counts of true negatives.
`data`	an optional data frame containing the meta-analysis dataset. If `data` is specified, the previous arguments, `tp`, `fp`, `fn`, and `tn`, should be specified as their corresponding column names in `data`.
`method`	a character string specifying the method used to implement the meta-analysis of diagnostic tests. It should be one of `"s.roc"` (summary ROC approach), `"biv.lmm"` (bivariate linear mixed model), and `"biv.glmm"` (bivariate generalized linear mixed model, the default). See details.
`alpha`	a numeric value specifying the statistical significance level.
`...`	other arguments that can be passed to the function `lm` (when `method` = `"s.roc"`), the function `rma.mv` in the package **metafor** (when `method` = `"biv.lmm"`), or the function `glmer` in the package **lme4** (when `method` = `"biv.glmm"`).

Details

Suppose a meta-analysis of diagnostic tests contains N studies. Each study reports the counts of true positives, false positives, false negatives, and true negatives, denoted by TP_i, FP_i, FN_i, and TN_i, respectively. The study-specific estimates of sensitivity and specificity are calculated as $Se_i = TP_i/(TP_i + FN_i)$ and $Sp_i = TN_i/(FP_i + TN_i)$ for $i = 1, \cdots, N$. They are analyzed on the logarithmic scale in the meta-analysis. When using the summary ROC (receiver operating characteristic) approach or the bivariate linear mixed model, 0.5 needs to be added to all four counts in a study when at least one count is zero.

The summary ROC approach first calculates

$$D_i = \log\left(\frac{Se_i}{1 - Se_i}\right) + \log\left(\frac{Sp_i}{1 - Sp_i}\right);$$

$$S_i = \log\left(\frac{Se_i}{1 - Se_i}\right) - \log\left(\frac{Sp_i}{1 - Sp_i}\right),$$

where D_i represents the log diagnostic odds ratio (DOR) in study i. A linear regression is then fitted:

$$D_i = \alpha + \beta \cdot S_i.$$

The regression could be either unweighted or weighted; this function performs both versions. If weighted, the study-specific weights are the inverse of the variances of D_i, i.e., $1/TP_i + 1/FP_i + 1/FN_i + 1/TN_i$. Based on the estimated regression intercept $\hat{\alpha}$ and slope $\hat{\beta}$, one may obtain the DOR at mean of S_i, Q point, summary ROC curve, and area under the curve (AUC). The Q point is the point on the summary ROC curve where sensitivity and specificity are equal. The ROC curve is given by

$$Se = \left\{1 + e^{-\hat{\alpha}/(1-\hat{\beta})} \cdot [Sp/(1-Sp)]^{(1+\hat{\beta})/(1-\hat{\beta})}\right\}^{-1}.$$

See more details of the summary ROC approach in Moses et al. (1993) and Irwig et al. (1995).

The bivariate linear mixed model described in Reitsma et al. (2005) assumes that the logit sensitivity and logit specificity independently follow normal distributions within each study: $g(Se_i) \sim N(\theta_{i,Se}, s^2_{i,Se})$ and $g(Sp_i) \sim N(\theta_{i,Sp}, s^2_{i,Sp})$, where $g(\cdot)$ denotes the logit function. The within-study variances are calculated as $s^2_{i,Se} = 1/TP_i + 1/FN_i$ and $s^2_{i,Sp} = 1/FP_i + 1/TN_i$. The parameters $\theta_{i,Se}$ and $\theta_{i,Sp}$ are the underlying true sensitivity and specificity (on the logit scale) in study i. They are assumed to be random effects, jointly following a bivariate normal distribution:

$$(\theta_{i,Se}, \theta_{i,Sp})' \sim N\left((\mu_{Se}, \mu_{Sp})', \mathbf{\Sigma}\right),$$

where $\mathbf{\Sigma}$ is the between-study variance-covariance matrix. The diagonal elements of $\mathbf{\Sigma}$ are σ^2_{Se} and σ^2_{Sp}, representing the heterogeneity variances of sensitivities and specificities (on the logit scale), respectively. The correlation coefficient is ρ.

The bivariate generalized linear mixed model described in Chu and Cole (2006) refines the bivariate linear mixed model by directly modeling the counts of true positives and true negatives. This approach does not require the assumption that the logit sensitivity and logit specificity approximately follow normal distributions within studies, which could be seriously violated in the presence of small data counts. It also avoids corrections for zero counts. Specificially, the counts of true positives and true negatives are modeled using binomial likelihoods:

$$TP_i \sim Bin(TP_i + FN_i, Se_i);$$

$$TN_i \sim Bin(FP_i + TN_i, Sp_i);$$

$$(g(Se_i), g(Sp_i))' \sim N\left((\mu_{Se}, \mu_{Sp})', \mathbf{\Sigma}\right).$$

See more details in Chu and Cole (2006) and Ma et al. (2016).

For both the bivariate linear mixed model and bivariate generalized linear mixed model, μ_{Se} and μ_{Sp} represent the overall sensitivity and specificity (on the logit scale) across studies, respectively, and $\mu_{Se} + \mu_{Sp}$ represents the log DOR. The summary ROC curve may be constructed as

$$Se = \left\{1 + e^{-\hat{\mu}_{Se} + \hat{\mu}_{Sp} \cdot \hat{\rho}\hat{\sigma}_{Se}/\hat{\sigma}_{Sp}} \cdot [Sp/(1-Sp)]^{-\rho\hat{\sigma}_{Se}/\hat{\sigma}_{Sp}}\right\}^{-1}.$$

Value

This function returns a list of the meta-analysis results. When `method = "s.roc"`, the list consists of the regression intercept (`inter.unwtd`), slope (`slope.unwtd`), their variance-covariance matrix (`vcov.unwtd`), DOR at mean of S_i (`DOR.meanS.unwtd`) with its confidence interval (`DOR.meanS.unwtd.ci`),

Q point (Q.unwtd) with its confidence interval (Q.unwtd.ci), and AUC (AUC.unwtd) for the unweighted regression; it also consists of the counterparts for the weighted regression. When method = "biv.lmm" or "biv.glmm", the list consists of the overall sensitivity (sens.overall) with its confidence interval (sens.overall.ci), overall specificity (spec.overall) with its confidence interval (spec.overall.ci), overall DOR (DOR.overall) with its confidence interval (DOR.overall.ci), AUC (AUC), estimated μ_{Se} (mu.sens), μ_{Sp} (mu.spec), their variance-covariance matrix (mu.vcov), estimated σ_{Se} (sig.sens), σ_{Sp} (sig.spec), and ρ (rho). In addition, the list includes the method used to perform the meta-analysis of diagnostic tests (method), significance level (alpha), and original data (data).

Note

The original articles by Reitsma et al. (2005) and Chu and Cole (2006) used SAS to implement (generalized) linear mixed models (specifically, PROC MIXED and PROC NLMIXED); this function imports rma.mv from the package **metafor** and glmer from the package **lme4** for implementing these models. The estimation approaches adopted in SAS and the R packages **metafor** and **lme4** may differ, which may impact the results. See, for example, Zhang et al. (2011).

Author(s)

Lifeng Lin, Kristine J. Rosenberger

References

Chu H, Cole SR (2006). "Bivariate meta-analysis of sensitivity and specificity with sparse data: a generalized linear mixed model approach." *Journal of Clinical Epidemiology*, **59**(12), 1331–1332. <doi:10.1016/j.jclinepi.2006.06.011>

Irwig L, Macaskill P, Glasziou P, Fahey M (1995). "Meta-analytic methods for diagnostic test accuracy." *Journal of Clinical Epidemiology*, **48**(1), 119–130. <doi:10.1016/08954356(94)00099-C>

Ma X, Nie L, Cole SR, Chu H (2016). "Statistical methods for multivariate meta-analysis of diagnostic tests: an overview and tutorial." *Statistical Methods in Medical Research*, **25**(4), 1596–1619. <doi:10.1177/0962280213492588>

Moses LE, Shapiro D, Littenberg B (1993). "Combining independent studies of a diagnostic test into a summary ROC curve: data-analytic approaches and some additional considerations." *Statistics in Medicine*, **12**(14), 1293–1316. <doi:10.1002/sim.4780121403>

Reitsma JB, Glas AS, Rutjes AWS, Scholten RJPM, Bossuyt PM, Zwinderman AH (2005). "Bivariate analysis of sensitivity and specificity produces informative summary measures in diagnostic reviews." *Journal of Clinical Epidemiology*, **58**(10), 982–990. <doi:10.1016/j.jclinepi.2005.02.022>

Zhang H, Lu N, Feng C, Thurston SW, Xia Y, Zhu L, Tu XM (2011). "On fitting generalized linear mixed-effects models for binary responses using different statistical packages." *Statistics in Medicine*, **30**(20), 2562–2572. <doi:10.1002/sim.4265>

See Also

maprop.twostep, meta.biv, plot.meta.dt, print.meta.dt

Examples

```
data("dat.scheidler")
out1 <- meta.dt(tp, fp, fn, tn, data = dat.scheidler[dat.scheidler$dt == "MRI",],
  method = "s.roc")
out1
plot(out1)
out2 <- meta.dt(tp, fp, fn, tn, data = dat.scheidler[dat.scheidler$dt == "MRI",],
  method = "biv.lmm")
out2
plot(out2, predict = TRUE)
out3 <- meta.dt(tp, fp, fn, tn, data = dat.scheidler[dat.scheidler$dt == "MRI",],
  method = "biv.glmm")
out3
plot(out3, add = TRUE, studies = FALSE,
  col.roc = "blue", col.overall = "blue", col.confid = "blue",
  predict = TRUE,col.predict = "blue")

data("dat.smith")
out4 <- meta.dt(tp, fp, fn, tn, data = dat.smith, method = "biv.glmm")
out4
plot(out4, predict = TRUE)
```

`meta.or.smd` *Meta-Analysis of Combining Standardized Mean Differences and Odds Ratios*

Description

Performs a Bayesian meta-analysis to synthesize standardized mean differences (SMDs) for a continuous outcome and odds ratios (ORs) for a binary outcome.

Usage

```
meta.or.smd(y1, sd1, n1, y0, sd0, n0, r1, m1, r0, m0, id.bin, data,
         n.adapt = 1000, n.chains = 3, n.burnin = 5000, n.iter = 20000, n.thin = 2,
            seed = 1234)
```

Arguments

`y1`	a vector specifying the sample means in the treatment group for the continuous outcome. `NA` is allowed and indicates that data are not available for this outcome; the same applies to the other arguments, including `sd1`, `n1`, `y0`, `sd0`, `n0`, `r1`, `m1`, `r0`, and `m0`.
`sd1`	a vector specifying the sample standard deviations in the treatment group for the continuous outcome.
`n1`	a vector specifying the sample sizes in the treatment group for the continuous outcome.

y0	a vector specifying the sample means in the control group for the continuous outcome.
sd0	a vector specifying the sample standard deviations in the control group for the continuous outcome.
n0	a vector specifying the sample sizes in the control group for the continuous outcome.
r1	a vector specifying the event counts in the treatment group for the binary outcome.
m1	a vector specifying the sample sizes in the treatment group for the binary outcome.
r0	a vector specifying the event counts in the control group for the binary outcome.
m0	a vector specifying the sample sizes in the control group for the binary outcome.
id.bin	a vector indicating whether the outcome is binary (1) or continuous (0).
data	an optional data frame containing the meta-analysis dataset. If data is specified, the previous arguments, y1, sd1, n1, y0, sd0, n0, r1, m1, r0, m0, and id.bin should be specified as their corresponding column names in data.
n.adapt	the number of iterations for adaptation in the Markov chain Monte Carlo (MCMC) algorithm. The default is 1,000. This argument and the following n.chains, n.burnin, n.iter, and n.thin are passed to the functions in the package **rjags**.
n.chains	the number of MCMC chains. The default is 3.
n.burnin	the number of iterations for burn-in period. The default is 5,000.
n.iter	the total number of iterations in each MCMC chain after the burn-in period. The default is 20,000.
n.thin	a positive integer specifying thinning rate. The default is 2.
seed	an integer for specifying the seed of the random number generation for reproducibility during the MCMC algorithm for performing the Bayesian meta-analysis model.

Details

The Bayesian meta-analysis model implemented by this function is detailed in Section 2.5 of Jing et al. (2023).

Value

"This function returns a list of Bayesian estimates, including posterior medians and 95% credible intervals (comprising the 2.5% and 97.5% posterior quantiles) for the overall SMD (d), the between-study standard deviation (tau), and the individual studies' SMDs (theta).

Author(s)

Yaqi Jing, Lifeng Lin

References

Jing Y, Murad MH, Lin L (2023). "A Bayesian model for combining standardized mean differences and odds ratios in the same meta-analysis." *Journal of Biopharmaceutical Statistics*, **33**(2), 167–190. <doi:10.1080/10543406.2022.2105345>

Examples

```
data("dat.dep")
out <- meta.or.smd(y1, sd1, n1, y0, sd0, n0, r1, m1, r0, m0, id.bin, data = dat.dep)
out
```

`meta.pen` *A penalization approach to random-effects meta-analysis*

Description

Performs the penalization methods introduced in Wang et al. (2022) to achieve a compromise between the common-effect and random-effects model.

Usage

```
meta.pen(y, s2, data, tuning.para = "tau", upp = 1, n.cand = 100, tol = 1e-10)
```

Arguments

`y`	a numeric vector or the corresponding column name in the argument data, specifying the observed effect sizes in the collected studies.
`s2`	a numeric vector or the corresponding column name in the argument data, specifying the within-study variances.
`data`	an optional data frame containing the meta-analysis dataset. If `data` is specified, the previous arguments, `y` and `s2`, should be specified as their corresponding column names in `data`.
`tuning.para`	a character string specifying the type of tuning parameter used in the penalization method. It should be one of "lambda" (use lambda as a tuning parameter) or "tau" (use the standard deviation as a tuning parameter). The default is "tau".
`upp`	a positive scalar used to control the upper bound of the range for the tuning parameter. Specifically, [0, T*upp] is used as the range of a tuning parameter. T is the upper threshold value of a tuning parameter. The default value of upp is 1.
`n.cand`	the total number of candidate values of the tuning parameter within the specified range. The default is 100.
`tol`	the desired accuracy (convergence tolerance). The default is 1e-10.

Details

Suppose a meta-analysis collects n independent studies. Let μ_i be the true effect size in study i (i = 1, ..., n). Each study reports an estimate of the effect size and its sample variance, denoted by y_i and s_i^2, respectively. These data are commonly modeled as y_i from $N(\mu_i, s_i^2)$. If study-specific true effect sizes μ_i are assumed i.i.d. from $N(\mu, \tau^2)$, this is the random-effects (RE) model, where μ is the overall effect size and τ^2 is the between-study variance. If $\tau^2 = 0$ and thus $\mu_i = \mu$ for all studies, this implies that studies are homogeneous and the RE model is reduced to the common-effect (CE) model.

Marginally, the RE model yields $y_i \sim N(\mu, s_i^2 + \tau^2)$, and its log-likelihood is

$$l(\mu, \tau^2) = -\frac{1}{2}\sum_{i=1}^{n}\left[\log(s_i^2 + \tau^2) + \frac{(y_i - \mu)^2}{s_i^2 + \tau^2}\right] + C,$$

where C is a constant. In the past two decades, penalization methods have been rapidly developed for variable selection in high-dimensional data analysis to control model complexity and reduce the variance of parameter estimates. Borrowing the idea from the penalization methods, we employ a penalty term on the between-study variance τ^2 when the heterogeneity is overestimated. The penalty term increases with τ^2. Specifically, we consider the following optimization problem:

$$(\hat{\mu}(\lambda), \hat{\tau}^2(\lambda)) = \min_{\mu, \tau^2 \geq 0}\left\{\sum_{i=1}^{n}\left[\log(s_i^2 + \tau^2) + \frac{(y_i - \mu)^2}{s_i^2 + \tau^2}\right] + \lambda\tau^2\right\},$$

where $\lambda \geq 0$ is a tuning parameter that controls the penalty strength. Using the technique of profile likelihood by taking the target function's derivative in the above equation with respect to μ for a given τ^2. The bivariate optimization problem is reduced to a univariate minimization problem. When $\lambda = 0$, the minimization problem is equivalent to minimizing the log-likelihood without penalty, so the penalized-likelihood method is identical to the conventional RE model. By contrast, it can be shown that a sufficiently large λ produces the estimated between-study variance as 0, leading to the conventional CE model.

As different tuning parameters lead to different estimates of $\hat{\mu}(\lambda)$ and $\hat{\tau}^2(\lambda)$, it is important to select the optimal λ among a set of candidate values. We perform the cross-validation process and construct a loss function of λ to measure the performance of specific λ values. The λ corresponding to the smallest loss is considered optimal. The threshold, denoted by $\lambda_{\max}$, based on the penalty function $p(\tau^2) = \tau^2$ can be calculated. For all $\lambda > \lambda_{\max}$, the estimated between-study variance is 0. Consequently, we select a certain number of candidate values (e.g., 100) from the range $[0, \lambda_{\max}]$ for the tuning parameter. For a set of tuning parameters, the leave-one-study-out (i.e., n-fold) cross-validation is used to construct the loss function. Specifically, we use the following loss function for the penalization method by tuning λ:

$$\hat{L}(\lambda) = \left[\frac{1}{n}\sum_{i=1}^{n}\frac{(y_i - \hat{\mu}_{(-i)}(\lambda))^2}{s_i^2 + \hat{\tau}^2_{RE(-i)} + \frac{\sum_{j\neq i}(s_j^2 + \hat{\tau}^2_{RE(-i)})/(s_j^2 + \hat{\tau}^2_{(-i)}(\lambda))^2}{(\sum_{j\neq i} 1/(s_j^2 + \hat{\tau}^2_{(-i)}(\lambda)))^2}}\right]^{1/2},$$

where the subscript $(-i)$ indicates that study i is removed, $\hat{\tau}^2_{(-i)}(\lambda)$ is the estimated between-study variance for a given λ, $\hat{\tau}^2_{RE(-i)}$ is the corresponding between-study variance estimate of the RE model, and

$$\hat{\mu}_{(-i)}(\lambda) = \frac{\sum_{j\neq i} y_j/(s_j^2 + \hat{\tau}^2_{(-i)}(\lambda))}{\sum_{j\neq i} 1/(s_j^2 + \hat{\tau}^2_{(-i)}(\lambda))}.$$

The above procedure focuses on tuning the parameter, λ, to control the penalty strength for the between-study variance. Alternatively, for the purpose of shrinking the potentially overestimated heterogeneity, we can directly treat the between-study standard deviation (SD), τ, as the tuning parameter. A set of candidate values of τ are considered, and the value that produces the minimum loss function is selected. Compared with tuning λ from the perspective of penalized likelihood, tuning τ is more straightforward and intuitive from the practical perspective. The candidate values of τ can be naturally chosen from $[0, \hat{\tau}_{RE}]$, with the lower and upper bounds corresponding to the CE and RE models, respectively. Denoted the candidate SDs as τ_t, the loss function with respect to τ_t is similarly defined as

$$\hat{L}(\tau_t) = \left[\frac{1}{n} \sum_{i=1}^{n} \frac{(y_i - \hat{\mu}_{(-i)}(\tau_t))^2}{s_i^2 + \hat{\tau}_{RE(-i)} + \frac{\sum_{j \neq i} (s_j^2 + \hat{\tau}^2_{RE(-i)})/(s_j^2 + \tau_t^2)^2}{[\sum_{j \neq i} 1/(s_j^2 + \tau_t^2)]^2}} \right]^{1/2},$$

where the overall effect size estimate (excluding study i) is

$$\hat{\mu}_{(-i)}(\tau_t) = \frac{\sum_{j \neq i} y_j/(s_j^2 + \tau_t^2)}{\sum_{j \neq i} 1/(s_j^2 + \tau_t^2)}.$$

Value

This function returns a list containing estimates of the overall effect size and their 95% confidence intervals. Specifically, the components include:

`n`	the number of studies in the meta-analysis.
`I2`	the I^2 statistic for quantifying heterogeneity.
`tau2.re`	the maximum likelihood estimate of the between-study variance.
`mu.fe`	the estimated overall effect size of the `CE` model.
`se.fe`	the standard deviation of the overall effect size estimate of the `CE` model.
`mu.re`	the estimated overall effect size of the `RE` model.
`se.re`	the standard deviation of the overall effect size estimate of the `RE` model.
`loss`	the values of the loss function for candidate tuning parameters.
`tau.cand`	the candidate values of the tuning parameter τ.
`lambda.cand`	the candidate values of the tuning parameter λ.
`tau.opt`	the estimated between-study standard deviation of the penalization method.
`mu.opt`	the estimated overall effect size of the penalization method.
`se.opt`	the standard error estimate of the estimated overall effect size of the penalization method.

Author(s)

Yipeng Wang <yipeng.wang1@ufl.edu>

References

Wang Y, Lin L, Thompson CG, Chu H (2022). "A penalization approach to random-effects meta-analysis." *Statistics in Medicine*, **41**(3), 500–516. <doi:10.1002/sim.9261>

Examples

```
data("dat.bohren") ## log odds ratio
## perform the penaliztion method by tuning tau
out11 <- meta.pen(y, s2, dat.bohren)
## plot the loss function and candidate taus
plot(out11$tau.cand, out11$loss, xlab = NA, ylab = NA,
  lwd = 1.5, type = "l", cex.lab = 0.8, cex.axis = 0.8)
title(xlab = expression(paste(tau[t])),
  ylab = expression(paste("Loss function ", ~hat(L)(tau[t]))),
  cex.lab = 0.8, line = 2)
idx <- which(out11$loss == min(out11$loss))
abline(v = out11$tau.cand[idx], col = "gray", lwd = 1.5, lty = 2)

## perform the penaliztion method by tuning lambda
out12 <- meta.pen(y, s2, dat.bohren, tuning.para = "lambda")
## plot the loss function and candidate lambdas
plot(log(out12$lambda.cand + 1), out12$loss, xlab = NA, ylab = NA,
  lwd=1.5, type = "l", cex.lab = 0.8, cex.axis = 0.8)
title(xlab = expression(log(lambda + 1)),
  ylab = expression(paste("Loss function ", ~hat(L)(lambda))),
  cex.lab = 0.8, line = 2)
idx <- which(out12$loss == min(out12$loss))
abline(v = log(out12$lambda.cand[idx] + 1), col = "gray", lwd = 1.5, lty = 2)


data("dat.bjelakovic") ## log odds ratio
## perform the penaliztion method by tuning tau
out21 <- meta.pen(y, s2, dat.bjelakovic)
## plot the loss function and candidate taus
plot(out21$tau.cand, out21$loss, xlab = NA, ylab = NA,
  lwd=1.5, type = "l", cex.lab = 0.8, cex.axis = 0.8)
title(xlab = expression(paste(tau[t])),
  ylab = expression(paste("Loss function ", ~hat(L)(tau[t]))),
  cex.lab = 0.8, line = 2)
idx <- which(out21$loss == min(out21$loss))
abline(v = out21$tau.cand[idx], col = "gray", lwd = 1.5, lty = 2)

out22 <- meta.pen(y, s2, dat.bjelakovic, tuning.para = "lambda")

data("dat.carless") ## log odds ratio
## perform the penaliztion method by tuning tau
out31 <- meta.pen(y, s2, dat.carless)
## plot the loss function and candidate taus
plot(out31$tau.cand, out31$loss, xlab = NA, ylab = NA,
  lwd=1.5, type = "l", cex.lab = 0.8, cex.axis = 0.8)
title(xlab = expression(paste(tau[t])),
  ylab = expression(paste("Loss function ", ~hat(L)(tau[t]))),
  cex.lab = 0.8, line = 2)
idx <- which(out31$loss == min(out31$loss))
abline(v = out31$tau.cand[idx], col = "gray", lwd = 1.5, lty = 2)

out32 <- meta.pen(y, s2, dat.carless, tuning.para = "lambda")
```

```
data("dat.adams") ## mean difference
out41 <- meta.pen(y, s2, dat.adams)
## plot the loss function and candidate taus
plot(out41$tau.cand, out41$loss, xlab = NA, ylab = NA,
  lwd=1.5, type = "l", cex.lab = 0.8, cex.axis = 0.8)
title(xlab = expression(paste(tau[t])),
  ylab = expression(paste("Loss function ", ~hat(L)(tau[t]))),
  cex.lab = 0.8, line = 2)
idx <- which(out41$loss == min(out41$loss))
abline(v = out41$tau.cand[idx], col = "gray", lwd = 1.5, lty = 2)

out42 <- meta.pen(y, s2, dat.adams, tuning.para = "lambda")
```

`metahet` *Meta-Analysis Heterogeneity Measures*

Description

Calculates various between-study heterogeneity measures in meta-analysis, including the conventional measures (e.g., I^2) and the alternative measures (e.g., I^2_r) which are robust to outlying studies; p-values of various tests are also calculated.

Usage

```
metahet(y, s2, data, n.resam = 1000)
```

Arguments

`y`	a numeric vector specifying the observed effect sizes in the collected studies; they are assumed to be normally distributed.
`s2`	a numeric vector specifying the within-study variances.
`data`	an optional data frame containing the meta-analysis dataset. If `data` is specified, the previous arguments, `y` and `s2`, should be specified as their corresponding column names in `data`.
`n.resam`	a positive integer specifying the number of resampling iterations for calculating p-values of test statistics and 95% confidence interval of heterogeneity measures.

Details

Suppose that a meta-analysis collects n studies. The observed effect size in study i is y_i and its within-study variance is s_i^2. Also, the inverse-variance weight is $w_i = 1/s_i^2$. The fixed-effect estimate of overall effect size is $\bar{\mu} = \sum_{i=1}^n w_i y_i / \sum_{i=1}^n w_i$. The conventional test statistic for heterogeneity is

$$Q = \sum_{i=1}^{n} w_i (y_i - \bar{\mu})^2.$$

Based on the Q statistic, the method-of-moments estimate of the between-study variance $\hat{\tau}^2_{DL}$ is (DerSimonian and Laird, 1986)

$$\hat{\tau}^2_{DL} = \max\left\{0, \frac{Q-(n-1)}{\sum_{i=1}^n w_i - \sum_{i=1}^n w_i^2/\sum_{i=1}^n w_i}\right\}.$$

Also, the H and I^2 statistics (Higgins and Thompson, 2002; Higgins et al., 2003) are widely used in practice because they do not depend on the number of collected studies n and the effect size scale; these two statistics are defined as

$$H = \sqrt{Q/(n-1)};$$
$$I^2 = \frac{Q-(n-1)}{Q}.$$

Specifically, the H statistic reflects the ratio of the standard deviation of the underlying mean from a random-effects meta-analysis compared to the standard deviation from a fixed-effect meta-analysis; the I^2 statistic describes the proportion of total variance across studies that is due to heterogeneity rather than sampling error.

Outliers are frequently present in meta-analyses, and they may have great impact on the above heterogeneity measures. Alternatively, to be more robust to outliers, the test statistic may be modified as (Lin et al., 2017):

$$Q_r = \sum_{i=1}^n \sqrt{w_i}|y_i - \bar{\mu}|.$$

Based on the Q_r statistic, the method-of-moments estimate of the between-study variance $\hat{\tau}^2_r$ is defined as the solution to

$$Q_r\sqrt{\frac{\pi}{2}} = \sum_{i=1}^n \left\{1 - \frac{w_i}{\sum_{j=1}^n w_j} + \tau^2\left[w_i - \frac{2w_i^2}{\sum_{j=1}^n w_j} + \frac{w_i\sum_{j=1}^n w_j^2}{(\sum_{j=1}^n w_j)^2}\right]\right\}.$$

If no positive solution exists to the equation above, set $\hat{\tau}^2_r = 0$. The counterparts of the H and I^2 statistics are defined as

$$H_r = Q_r\sqrt{\pi/[2n(n-1)]};$$
$$I^2_r = \frac{Q_r^2 - 2n(n-1)/\pi}{Q_r^2}.$$

To further improve the robustness of heterogeneity assessment, the weighted *mean* in the Q_r statistic may be replaced by the weighted *median* $\hat{\mu}_m$, which is the solution to $\sum_{i=1}^n w_i[I(\theta \geq y_i) - 0.5] = 0$ with respect to θ. The new test statistic is

$$Q_m = \sum_{i=1}^n \sqrt{w_i}|y_i - \hat{\mu}_m|.$$

Based on Q_m, the new estimator of the between-study variance $\hat{\tau}^2_m$ is the solution to

$$Q_m\sqrt{\pi/2} = \sum_{i=1}^n \sqrt{(s_i^2 + \tau^2)/s_i^2}.$$

The counterparts of the H and I^2 statistics are

$$H_m = \frac{Q_m}{n}\sqrt{\pi/2};$$
$$I^2_m = \frac{Q_m^2 - 2n^2/\pi}{Q_m^2}.$$

Value

This function returns a list containing p-values of various heterogeneity tests and various heterogeneity measures with 95% confidence intervals. Specifically, the components include:

`p.Q`	p-value of the Q statistic (using the resampling method).
`p.Q.theo`	p-value of the Q statistic using the Q's theoretical chi-squared distribution.
`p.Qr`	p-value of the Q_r statistic (using the resampling method).
`p.Qm`	p-value of the Q_m statistic (using the resampling method).
`Q`	the Q statistic.
`ci.Q`	95% CI of the Q statistic.
`tau2.DL`	DerSimonian–Laird estimate of the between-study variance.
`ci.tau2.DL`	95% CI of the between-study variance based on the DerSimonian–Laird method.
`H`	the H statistic.
`ci.H`	95% CI of the H statistic.
`I2`	the I^2 statistic.
`ci.I2`	95% CI of the I^2 statistic.
`Qr`	the Q_r statistic.
`ci.Qr`	95% CI of the Q_r statistic.
`tau2.r`	the between-study variance estimate based on the Q_r statistic.
`ci.tau2.r`	95% CI of the between-study variance based on the Q_r statistic.
`Hr`	the H_r statistic.
`ci.Hr`	95% CI of the H_r statistic.
`Ir2`	the I_r^2 statistic.
`ci.Ir2`	95% CI of the I_r^2 statistic.
`Qm`	the Q_m statistic.
`ci.Qm`	95% CI of the Q_m statistic.
`tau2.m`	the between-study variance estimate based on the Q_m statistic.
`ci.tau2.m`	95% CI of the between-study variance based on the Q_m statistic
`Hm`	the H_m statistic.
`ci.Hm`	95% CI of the H_m statistic.
`Im2`	the I_m^2 statistic.
`ci.Im2`	95% CI of the I_m^2 statistic.

References

DerSimonian R, Laird N (1986). "Meta-analysis in clinical trials." *Controlled Clinical Trials*, **7**(3), 177–188. <doi:10.1016/01972456(86)900462>

Higgins JPT, Thompson SG (2002). "Quantifying heterogeneity in a meta-analysis." *Statistics in Medicine*, **21**(11), 1539–1558. <doi:10.1002/sim.1186>

Higgins JPT, Thompson SG, Deeks JJ, Altman DG (2003). "Measuring inconsistency in meta-analyses." *BMJ*, **327**(7414), 557–560. <doi:10.1136/bmj.327.7414.557>

Lin L, Chu H, Hodges JS (2017). "Alternative measures of between-study heterogeneity in meta-analysis: reducing the impact of outlying studies." *Biometrics*, **73**(1), 156–166. <doi:10.1111/biom.12543>

Examples

```
data("dat.aex")
set.seed(1234)
metahet(y, s2, dat.aex, 100)
metahet(y, s2, dat.aex, 1000)

data("dat.hipfrac")
set.seed(1234)
metahet(y, s2, dat.hipfrac, 100)
metahet(y, s2, dat.hipfrac, 1000)
```

`metahet.hybrid` *Alternative Tests and Measures for Between-Study Inconsistency in Meta-Analysis*

Description

Performs several alternative tests (including a hybrid test) and calculates the associated measures for assessing between-study inconsistency in meta-analysis.

Usage

```
metahet.hybrid(y, s2, data, gam, testinf = TRUE, iter.resam = 200)
```

Arguments

`y`	Numeric vector of observed effect estimates, or the name of a column in `data`.
`s2`	Numeric vector of within-study variances corresponding to `y`, or the name of a column in `data`.
`data`	Optional data frame containing the variables `y` and `s2`.
`gam`	Numeric vector of power parameters γ, e.g., `1:8`.
`testinf`	Logical; if `TRUE`, includes the Q_∞ statistic (maximum standardized deviation).
`iter.resam`	Integer; number of bootstrap resamples used to estimate expected values and p-values.

Details

The function generalizes the adaptive sum of powered score (aSPU) framework by combining standardized powered deviations across studies. Specifically, for each power parameter gamma, it calculates a statistic that measures the average standardized deviation of individual study effects from the overall mean, with greater gamma values placing more weight on extreme deviations.

For each gamma, the expected value of this statistic is obtained through a parametric bootstrap under the null hypothesis of homogeneity. The observed statistic is then standardized by subtracting its expected value and dividing by the observed total deviation, resulting in the expected gamma (E-gamma) measure.

In addition, a hybrid test is computed by combining the p-values across all gamma values using the minimum-p combination rule. It provides an overall summary of between-study inconsistency that integrates evidence from multiple power levels.

Value

A list named `out` containing:

- `E_gamma`: Vector of E-gamma statistics for each γ, including E_∞ and the hybrid E_{Lhyb} term.
- `p_gamma`: Bootstrap p-values corresponding to each γ.
- `p_hyb`: Hybrid p-value combining all γ values.
- `gam`: Vector of power parameters used in the tests.
- `table`: Combined summary table for reporting.

Author(s)

Zhiyuan Yu, Xing Xing, Lifeng Lin

References

Pan W, Kim J, Zhang Y, Shen X, Wei P (2014). "A powerful and adaptive association test for rare variants." *Genetics*, **197**(4), 1081–1095. <doi:10.1534/genetics.114.165035>

Xu G, Lin L, Wei P, Pan W (2016). "An adaptive two-sample test for high-dimensional means." *Biometrika*, **103**(3), 609–624. <doi:10.1093/biomet/asw029>

Examples

```
data("dat.hughes")
set.seed(123)
## increase iter.resam to reduce Monte Carlo error due to resampling
metahet.hybrid(y, s2, data = dat.hughes, gam = 1:8,
  testinf = TRUE, iter.resam = 200)
```

`metaoutliers` *Outlier Detection in Meta-Analysis*

Description

Calculates the standardized residual for each study in meta-analysis using the methods desribed in Chapter 12 in Hedges and Olkin (1985) and Viechtbauer and Cheung (2010). A study is considered as an outlier if its standardized residual is greater than 3 in absolute magnitude.

Usage

```
metaoutliers(y, s2, data, model)
```

Arguments

`y`	a numeric vector specifying the observed effect sizes in the collected studies; they are assumed to be normally distributed.
`s2`	a numeric vector specifying the within-study variances.
`data`	an optional data frame containing the meta-analysis dataset. If `data` is specified, the previous arguments, `y` and `s2`, should be specified as their corresponding column names in `data`.
`model`	a character string specified as either `"FE"` or `"RE"`. If `model = "FE"`, this function uses the outlier detection procedure for the fixed-effect meta-analysis desribed in Chapter 12 in Hedges and Olkin (1985); If `model = "RE"`, the procedure for the random-effects meta-analysis desribed in Viechtbauer and Cheung (2010) is used. See Details for the two approaches. If the argument `model` is not specified, this function sets `model = "FE"` if $I_r^2 < 30\%$ and sets `model = "RE"` if $I_r^2 \geq 30\%$.

Details

Suppose that a meta-analysis collects n studies. The observed effect size in study i is y_i and its within-study variance is s_i^2. Also, the inverse-variance weight is $w_i = 1/s_i^2$.

Chapter 12 in Hedges and Olkin (1985) describes the outlier detection procedure for the fixed-effect meta-analysis (`model = "FE"`). Using the studies except study i, the pooled estimate of the overall effect size is $\bar{\mu}_{(-i)} = \sum_{j \neq i} w_j y_j / \sum_{j \neq i} w_j$. The residual of study i is $e_i = y_i - \bar{\mu}_{(-i)}$. The variance of e_i is $v_i = s_i^2 + (\sum_{j \neq i} w_j)^{-1}$, so the standardized residual of study i is $\epsilon_i = e_i / \sqrt{v_i}$.

Viechtbauer and Cheung (2010) describes the outlier detection procedure for the random-effects meta-analysis (`model = "RE"`). Using the studies except study i, let the method-of-moments estimate of the between-study variance be $\hat{\tau}^2_{(-i)}$. The pooled estimate of the overall effect size is $\bar{\mu}_{(-i)} = \sum_{j \neq i} \tilde{w}_{(-i)j} y_j / \sum_{j \neq i} \tilde{w}_{(-i)j}$, where $\tilde{w}_{(-i)j} = 1/(s_j^2 + \hat{\tau}^2_{(-i)})$. The residual of study i is $e_i = y_i - \bar{\mu}_{(-i)}$, and its variance is $v_i = s_i^2 + \hat{\tau}^2_{(-i)} + (\sum_{j \neq i} \tilde{w}_{(-i)j})^{-1}$. Then, the standardized residual of study i is $\epsilon_i = e_i / \sqrt{v_i}$.

Value

This functions returns a list which contains standardized residuals and identified outliers. A study is considered as an outlier if its standardized residual is greater than 3 in absolute magnitude.

References

Hedges LV, Olkin I (1985). *Statistical Method for Meta-Analysis*. Academic Press, Orlando, FL.

Viechtbauer W, Cheung MWL (2010). "Outlier and influence diagnostics for meta-analysis." *Research Synthesis Methods*, **1**(2), 112–125. <doi:10.1002/jrsm.11>

Examples

```
data("dat.aex")
metaoutliers(y, s2, dat.aex, model = "FE")
metaoutliers(y, s2, dat.aex, model = "RE")

data("dat.hipfrac")
metaoutliers(y, s2, dat.hipfrac)
```

---

metapb *Detecting and Quantifying Publication Bias/Small-Study Effects*

Description

Performs the regression test and calculates skewness for detecting and quantifying publication bias/small-study effects.

Usage

```
metapb(y, s2, data, model = "RE", n.resam = 1000)
```

Arguments

`y`	a numeric vector specifying the observed effect sizes in the collected studies; they are assumed to be normally distributed.
`s2`	a numeric vector specifying the within-study variances.
`data`	an optional data frame containing the meta-analysis dataset. If `data` is specified, the previous arguments, `y` and `s2`, should be specified as their corresponding column names in `data`.
`model`	a characher string specifying the fixed-effect (`"FE"`) or random-effects (`"RE"`, the default) model. If not specified, this function uses the Q statistic to test for heterogeneity: if the p-value is smaller than 0.05, `model` is set to `"RE"`; otherwise, `model` = `"FE"`.
`n.resam`	a positive integer specifying the number of resampling iterations.

Details

This function derives the measures of publication bias introduced in Lin and Chu (2018).

Value

This function returns a list containing measures of publication bias, their 95% confidence intervals, and p-values. Specifically, the components include:

`n`	the number of studies in the meta-analysis.
`p.Q`	the p-value of the Q-test for heterogeneity.
`I2`	the I^2 statistic for quantifying heterogeneity.
`tau2`	the DerSimonian–Laird estimate of the between-study variance.
`model`	the model setting (`"FE"` or `"RE"`).
`std.dev`	the standardized deviates of the studies.
`reg.int`	the estimate of the regression intercept for quantifying publication bias.
`reg.int.ci`	the 95% CI of the regression intercept.
`reg.int.ci.resam`	the 95% CI of the regression intercept based on the resampling method.
`reg.pval`	the p-value of the regression intercept.
`reg.pval`	the p-value of the regression intercept based on the resampling method.
`skewness`	the estimate of the skewness for quantifying publication bias.
`skewness.ci`	the 95% CI of the skewness.
`skewness.ci.resam`	the 95% CI of the skewness based on the resampling method.
`skewness.pval`	the p-value of the skewness.
`skewness.pval.resam`	the p-value of the skewness based on the resampling method.
`combined.pval`	the p-value of the combined test that incorporates the regression intercept and the skewness.
`combined.pval.resam`	the p-value of the combined test that incorporates the regression intercept and the skewness based on the resampling method.

References

Egger M, Davey Smith G, Schneider M, Minder C (1997). "Bias in meta-analysis detected by a simple, graphical test." *BMJ*, **315**(7109), 629–634. <doi:10.1136/bmj.315.7109.629>

Lin L, Chu H (2018). "Quantifying publication bias in meta-analysis." *Biometrics*, **74**(3), 785–794. <doi:10.1111/biom.12817>

Examples

```
data("dat.slf")
set.seed(1234)
metapb(y, s2, dat.slf)

data("dat.ha")
set.seed(1234)
metapb(y, s2, dat.ha)

data("dat.lcj")
set.seed(1234)
metapb(y, s2, dat.lcj)
```

---

mvma *Multivariate Meta-Analysis*

Description

Performs a multivariate meta-analysis when the within-study correlations are known.

Usage

```
mvma(ys, covs, data, method = "reml", tol = 1e-10)
```

Arguments

ys	an n x p numeric matrix containing the observed effect sizes. The n rows represent studies, and the p columns represent the multivariate endpoints. NA is allowed for missing endpoints.
covs	a numeric list with length n. Each element is the p x p within-study covariance matrix. NA is allowed for missing endpoints in the covariance matrix.
data	an optional data frame containing the multivariate meta-analysis dataset. If data is specified, the previous arguments, ys and covs, should be specified as their corresponding column names in data.
method	a character string specifying the method for estimating the overall effect sizes. It should be "fe" (fixed-effects model), "ml" (random-effects model using the maximum likelihood method), or "reml" (random-effects model using the restricted maximum likelihood method, the default).
tol	a small number specifying the convergence tolerance for the estimates by maximizing (restricted) likelihood. The default is 1e-10.

Details

Suppose n studies are collected in a multivariate meta-analysis on a total of p endpoints. Denote the p-dimensional vector of effect sizes as $\boldsymbol{y}_i$, and the within-study covariance matrix $\mathbf{S}_i$ is assumed to be known. Then, the random-effects model is as follows:

$$\boldsymbol{y}_i \sim N(\boldsymbol{\mu}_i, \mathbf{S}_i);$$

$$\boldsymbol{\mu}_i \sim N(\boldsymbol{\mu}, \mathbf{T}).$$

Here, $\boldsymbol{\mu}_i$ represents the true underlying effect sizes in study i, $\boldsymbol{\mu}$ represents the overall effect sizes across studies, and $\mathbf{T}$ is the between-study covariance matrix due to heterogeneity. By setting $\mathbf{T} = \mathbf{0}$, this model becomes the fixed-effects model.

Value

This function returns a list containing the following elements:

`mu.est`	The estimated overall effect sizes of the p endpoints.
`Tau.est`	The estimated between-study covariance matrix.
`mu.cov`	The covariance matrix of the estimated overall effect sizes.
`method`	The method used to produce the estimates.

References

Jackson D, Riley R, White IR (2011). "Multivariate meta-analysis: potential and promise." *Statistics in Medicine*, **30**(20), 2481–2498. <doi:10.1002/sim.4172>

See Also

`mvma.bayesian`, `mvma.hybrid`, `mvma.hybrid.bayesian`

Examples

```
data("dat.fib")
mvma(ys = y, covs = S, data = dat.fib, method = "fe")

mvma(ys = y, covs = S, data = dat.fib, method = "reml")
```

`mvma.bayesian` *Bayesian Random-Effects Multivariate Meta-Analysis*

Description

Performs a Bayesian random-effects model for multivariate meta-analysis when the within-study correlations are known.

Usage

```
mvma.bayesian(ys, covs, data, n.adapt = 1000, n.chains = 3,
              n.burnin = 10000, n.iter = 10000, n.thin = 1,
              data.name = NULL, traceplot = FALSE, coda = FALSE)
```

Arguments

ys	an n x p numeric matrix containing the observed effect sizes. The n rows represent studies, and the p columns represent the multivariate endpoints. NA is allowed for missing endpoints.
covs	a numeric list with length n. Each element is the p x p within-study covariance matrix. NA is allowed for missing endpoints in the covariance matrix.
data	an optional data frame containing the multivariate meta-analysis dataset. If data is specified, the previous arguments, ys and covs, should be specified as their corresponding column names in data.
n.adapt	the number of iterations for adaptation in the Markov chain Monte Carlo (MCMC) algorithm. The default is 1,000. This argument and the following n.chains, n.burnin, n.iter, and n.thin are passed to the functions in the package **rjags**.
n.chains	the number of MCMC chains. The default is 3.
n.burnin	the number of iterations for burn-in period. The default is 10,000.
n.iter	the total number of iterations in each MCMC chain after the burn-in period. The default is 10,000.
n.thin	a positive integer specifying thinning rate. The default is 1.
data.name	a character string specifying the data name. This is used in the names of the generated files that contain results. The default is NULL.
traceplot	a logical value indicating whether to save trace plots for the overall effect sizes and between-study standard deviations. The default is FALSE.
coda	a logical value indicating whether to output MCMC posterior samples. The default is FALSE.

Details

Suppose n studies are collected in a multivariate meta-analysis on a total of p endpoints. Denote the p-dimensional vector of effect sizes as $\boldsymbol{y}_i$, and the within-study covariance matrix $\mathbf{S}_i$ is assumed to be known. Then, the random-effects model is as follows:

$$\boldsymbol{y}_i \sim N(\boldsymbol{\mu}_i, \mathbf{S}_i);$$

$$\boldsymbol{\mu}_i \sim N(\boldsymbol{\mu}, \mathbf{T}).$$

Here, $\boldsymbol{\mu}_i$ represents the true underlying effect sizes in study i, $\boldsymbol{\mu}$ represents the overall effect sizes across studies, and $\mathbf{T}$ is the between-study covariance matrix due to heterogeneity.

The vague priors $N(0, 10^3)$ are specified for the fixed effects $\boldsymbol{\mu}$. Also, this function uses the separation strategy to specify vague priors for the variance and correlation components in $\mathbf{T}$ (Pinheiro and Bates, 1996); this technique is considered less sensitive to hyperparameters compared

to specifying the inverse-Wishart prior (Lu and Ades, 2009; Wei and Higgins, 2013). Specifically, write the between-study covariance matrix as $\mathbf{T} = \mathbf{D}^{1/2}\mathbf{R}\mathbf{D}^{1/2}$, where the diagonal matrix $\mathbf{D} = diag(\mathbf{T}) = diag(\tau_1^2, \ldots, \tau_p^2)$ contains the between-study variances, and $\mathbf{R}$ is the correlation matrix. Uniform priors $U(0, 10)$ are specified for τ_j's ($j = 1, \ldots, p$). Further, the correlation matrix can be written as $\mathbf{R} = \mathbf{L}\mathbf{L}'$, where $\mathbf{L} = (L_{ij})$ is a lower triangular matrix with nonnegative diagonal elements. Also, $L_{11} = 1$ and for $i = 2, \ldots, p$, $L_{ij} = \cos\theta_{i2}$ if $j = 1$; $L_{ij} = (\prod_{k=2}^{j} \sin\theta_{ik}) \cos\theta_{i,j+1}$ if $j = 2, \ldots, i-1$; and $L_{ij} = \prod_{k=2}^{i} \sin\theta_{ik}$ if $j = i$. Here, θ_{ij}'s are angle parameters for $2 \leq j \leq i \leq p$, and $\theta_{ij} \in (0, \pi)$. Uniform priors are specified for the angle parameters: $\theta_{ij} \sim U(0, \pi)$.

Value

This functions produces posterior estimates and Gelman and Rubin's potential scale reduction factor, and it generates several files that contain trace plots (if `traceplot = TRUE`) and MCMC posterior samples (if `coda = TRUE`) in users' working directory. In these results, `mu` represents the overall effect sizes, `tau` represents the between-study variances, `R` contains the elements of the correlation matrix, and `theta` represents the angle parameters (see "Details").

Note

This function only implements the MCMC algorithm for the random-effects multivariate model, but not the fixed-effects model. Generally, the fixed-effects model can be easily implemented using the function `mvma`. However, when using `mvma` to fit the random-effects model, a large number of parameters need to be estimated, and the algorithm for maximizing (restricted) likelihood may not converge well. The Bayesian method in this function provides an alternative.

If a warning "adaptation incomplete" appears, users may increase `n.adapt`.

References

Gelman A, Rubin DB (1992). "Inference from iterative simulation using multiple sequences." *Statistical Science*, **7**(4), 457–472. <doi:10.1214/ss/1177011136>

Jackson D, Riley R, White IR (2011). "Multivariate meta-analysis: potential and promise." *Statistics in Medicine*, **30**(20), 2481–2498. <doi:10.1002/sim.4172>

Lu G, Ades AE (2009). "Modeling between-trial variance structure in mixed treatment comparisons." *Biostatistics*, **10**(4), 792–805. <doi:10.1093/biostatistics/kxp032>

Pinheiro JC, Bates DM (1996). "Unconstrained parametrizations for variance-covariance matrices." *Statistics and Computing*, **6**(3), 289–296. <doi:10.1007/BF00140873>

Wei Y, Higgins JPT (2013). "Bayesian multivariate meta-analysis with multiple outcomes." *Statistics in Medicine*, **32**(17), 2911–2934. <doi:10.1002/sim.5745>

See Also

`mvma`, `mvma.hybrid`, `mvma.hybrid.bayesian`

Examples

```
data("dat.fib")
set.seed(12345)
```

```
## increase n.burnin and n.iter for better convergence of MCMC
out <- mvma.bayesian(ys = y, covs = S, data = dat.fib,
  n.adapt = 1000, n.chains = 3, n.burnin = 100, n.iter = 100,
  n.thin = 1, data.name = "Fibrinogen")
out
```

`mvma.hybrid` *Hybrid Model for Random-Effects Multivariate Meta-Analysis*

Description

Performs a multivariate meta-analysis using the hybrid random-effects model when the within-study correlations are unknown.

Usage

```
mvma.hybrid(ys, vars, data, method = "reml", tol = 1e-10)
```

Arguments

`ys`	an n x p numeric matrix containing the observed effect sizes. The n rows represent studies, and the p columns represent the multivariate endpoints. `NA` is allowed for missing endpoints.
`vars`	an n x p numeric matrix containing the observed within-study variances. The n rows represent studies, and the p columns represent the multivariate endpoints. `NA` is allowed for missing endpoints.
`data`	an optional data frame containing the multivariate meta-analysis dataset. If `data` is specified, the previous arguments, `ys` and `vars`, should be specified as their corresponding column names in `data`.
`method`	a character string specifying the method for estimating the overall effect sizes. It should be `"ml"` (random-effects model using the maximum likelihood method) or `"reml"` (random-effects model using the restricted maximum likelihood method, the default).
`tol`	a small number specifying the convergence tolerance for the estimates by maximizing (restricted) likelihood. The default is `1e-10`.

Details

Suppose n studies are collected in a multivariate meta-analysis on a total of p endpoints. Denote the p-dimensional vector of effect sizes as $\boldsymbol{y}_i$, and their within-study variances form a diagonal matrix $\mathbf{D}_i$. However, the within-study correlations are unknown. Then, the random-effects hybrid model is as follows (Riley et al., 2008; Lin and Chu, 2018):

$$\boldsymbol{y}_i \sim N(\boldsymbol{\mu}, (\mathbf{D}_i + \mathbf{T})^{1/2}\mathbf{R}(\mathbf{D}_i + \mathbf{T})^{1/2}),$$

where $\boldsymbol{\mu}$ represents the overall effect sizes across studies, $\mathbf{T} = diag(\tau_1^2, \ldots, \tau_p^2)$ consists of the between-study variances, and $\mathbf{R}$ is the marginal correlation matrix. Although the within-study correlations are unknown, this model accounts for both within- and between-study correlations by using the marginal correlation matrix.

Value

This function returns a list containing the following elements:

`mu.est`	The estimated overall effect sizes of the p endpoints.
`tau2.est`	The estimated between-study variances of the p endpoints.
`mar.R`	The estimated marginal correlation matrix.
`mu.cov`	The covariance matrix of the estimated overall effect sizes.
`method`	The method used to produce the estimates.

Note

The algorithm for maximizing (restricted) likelihood may not converge when the dimension of endpoints is too high or the data are too sparse.

References

Lin L, Chu H (2018), "Bayesian multivariate meta-analysis of multiple factors." *Research Synthesis Methods*, **9**(2), 261–272. <doi:10.1002/jrsm.1293>

Riley RD, Thompson JR, Abrams KR (2008), "An alternative model for bivariate random-effects meta-analysis when the within-study correlations are unknown." *Biostatistics*, **9**(1), 172–186. <doi:10.1093/biostatistics/kxm023>

See Also

`mvma`, `mvma.bayesian`, `mvma.hybrid.bayesian`

Examples

```
data("dat.fib")
y <- dat.fib$y
sd <- dat.fib$sd
mvma.hybrid(y = y, vars = sd^2)
```

`mvma.hybrid.bayesian` *Bayesian Hybrid Model for Random-Effects Multivariate Meta-Analysis*

Description

Performs a multivariate meta-analysis using the Bayesian hybrid random-effects model when the within-study correlations are unknown.

Usage

```
mvma.hybrid.bayesian(ys, vars, data, n.adapt = 1000, n.chains = 3,
                     n.burnin = 10000, n.iter = 10000, n.thin = 1,
                     data.name = NULL, traceplot = FALSE, coda = FALSE)
```

Arguments

ys	an n x p numeric matrix containing the observed effect sizes. The n rows represent studies, and the p columns represent the multivariate endpoints. NA is allowed for missing endpoints.
vars	an n x p numeric matrix containing the observed within-study variances. The n rows represent studies, and the p columns represent the multivariate endpoints. NA is allowed for missing endpoints.
data	an optional data frame containing the multivariate meta-analysis dataset. If data is specified, the previous arguments, ys and vars, should be specified as their corresponding column names in data.
n.adapt	the number of iterations for adaptation in the Markov chain Monte Carlo (MCMC) algorithm. The default is 1,000. This argument and the following n.chains, n.burnin, n.iter, and n.thin are passed to the functions in the package **rjags**.
n.chains	the number of MCMC chains. The default is 3.
n.burnin	the number of iterations for burn-in period. The default is 10,000.
n.iter	the total number of iterations in each MCMC chain after the burn-in period. The default is 10,000.
n.thin	a positive integer specifying thinning rate. The default is 1.
data.name	a character string specifying the data name. This is used in the names of the generated files that contain results. The default is NULL.
traceplot	a logical value indicating whether to save trace plots for the overall effect sizes and between-study standard deviations. The default is FALSE.
coda	a logical value indicating whether to output MCMC posterior samples. The default is FALSE.

Details

Suppose n studies are collected in a multivariate meta-analysis on a total of p endpoints. Denote the p-dimensional vector of effect sizes as $\boldsymbol{y}_i$, and their within-study variances form a diagonal matrix $\mathbf{D}_i$. However, the within-study correlations are unknown. Then, the random-effects hybrid model is as follows (Riley et al., 2008; Lin and Chu, 2018):

$$\boldsymbol{y}_i \sim N(\boldsymbol{\mu}, (\mathbf{D}_i + \mathbf{T})^{1/2}\mathbf{R}(\mathbf{D}_i + \mathbf{T})^{1/2}),$$

where $\boldsymbol{\mu}$ represents the overall effect sizes across studies, $\mathbf{T} = diag(\tau_1^2, \ldots, \tau_p^2)$ consists of the between-study variances, and $\mathbf{R}$ is the marginal correlation matrix. Although the within-study correlations are unknown, this model accounts for both within- and between-study correlations by using the marginal correlation matrix.

Uniform priors $U(0, 10)$ are specified for the between-study standard deviations τ_j ($j = 1, \ldots, p$). The correlation matrix can be written as $\mathbf{R} = \mathbf{L}\mathbf{L}'$, where $\mathbf{L} = (L_{ij})$ is a lower triangular matrix with nonnegative diagonal elements. Also, $L_{11} = 1$ and for $i = 2, \ldots, p$, $L_{ij} = \cos\theta_{i2}$ if $j = 1$; $L_{ij} = (\prod_{k=2}^{j} \sin\theta_{ik}) \cos\theta_{i,j+1}$ if $j = 2, \ldots, i-1$; and $L_{ij} = \prod_{k=2}^{i} \sin\theta_{ik}$ if $j = i$ (Lu and Ades, 2009; Wei and Higgins, 2013). Here, θ_{ij}'s are angle parameters for $2 \leq j \leq i \leq p$, and $\theta_{ij} \in (0, \pi)$. Uniform priors are specified for the angle parameters: $\theta_{ij} \sim U(0, \pi)$.

Value

This functions produces posterior estimates and Gelman and Rubin's potential scale reduction factor, and it generates several files that contain trace plots (if `traceplot = TRUE`), and MCMC posterior samples (if `coda = TRUE`) in users' working directory. In these results, `mu` represents the overall effect sizes, `tau` represents the between-study variances, `R` contains the elements of the correlation matrix, and `theta` represents the angle parameters (see "Details").

References

Lin L, Chu H (2018), "Bayesian multivariate meta-analysis of multiple factors." *Research Synthesis Methods*, **9**(2), 261–272. <doi:10.1002/jrsm.1293>

Lu G, Ades AE (2009). "Modeling between-trial variance structure in mixed treatment comparisons." *Biostatistics*, **10**(4), 792–805. <doi:10.1093/biostatistics/kxp032>

Riley RD, Thompson JR, Abrams KR (2008), "An alternative model for bivariate random-effects meta-analysis when the within-study correlations are unknown." *Biostatistics*, **9**(1), 172–186. <doi:10.1093/biostatistics/kxm023>

Wei Y, Higgins JPT (2013). "Bayesian multivariate meta-analysis with multiple outcomes." *Statistics in Medicine*, **32**(17), 2911–2934. <doi:10.1002/sim.5745>

See Also

`mvma`, `mvma.bayesian`, `mvma.hybrid`

Examples

```
data("dat.pte")
set.seed(12345)
## increase n.burnin and n.iter for better convergence of MCMC
out <- mvma.hybrid.bayesian(ys = dat.pte$y, vars = (dat.pte$se)^2,
  n.adapt = 1000, n.chains = 3, n.burnin = 100, n.iter = 100,
  n.thin = 1, data.name = "Pterygium")
out
```

`nma.icdf` *Evidence Inconsistency Degrees of Freedom in Bayesian Network Meta-Analysis*

Description

Calculates evidence inconsistency degrees of freedom (ICDF) in Bayesian network meta-analysis with binary outcomes.

Usage

```
nma.icdf(sid, tid, r, n, data, type = c("la", "fe", "re"),
         n.adapt = 1000, n.chains = 3, n.burnin = 5000, n.iter = 20000,
         n.thin = 2, traceplot = FALSE, nma.name = NULL, seed = 1234)
```

Arguments

sid	a vector specifying the study IDs.
tid	a vector specifying the treatment IDs.
r	a numeric vector specifying the event counts.
n	a numeric vector specifying the sample sizes.
data	an optional data frame containing the network meta-analysis dataset. If data is specified, the previous arguments, sid, tid, r, and n, should be specified as their corresponding column names in data.
type	a character string or a vector of character strings specifying the ICDF measures. It can be chosen from "la" (the Lu–Ades measure), "fe" (based on fixed-effects models), and "re" (based on random-effects models).
n.adapt	the number of iterations for adaptation in the Markov chain Monte Carlo (MCMC) algorithm. The default is 1,000. This argument and the following n.chains, n.burnin, n.iter, and n.thin are passed to the functions in the package **rjags**.
n.chains	the number of MCMC chains. The default is 3.
n.burnin	the number of iterations for burn-in period. The default is 5,000.
n.iter	the total number of iterations in each MCMC chain after the burn-in period. The default is 20,000.
n.thin	a positive integer specifying thinning rate. The default is 2.
traceplot	a logical value indicating whether to generate trace plots of network meta-analysis models for calculating the ICDF measures. It is only used when the argument type includes "fe" and/or "re".
nma.name	a character string for specifying the name of the network meta-analysis, which will be used in the file names of the trace plots. It is only used when traceplot = TRUE.
seed	an integer for specifying the seed of the random number generation for reproducibility during the MCMC algorithm for performing Bayesian network meta-analysis models.

Details

Network meta-analysis frequenlty assumes that direct evidence is consistent with indirect evidence, but this consistency assumption may not hold in some cases. One may use the ICDF to assess the potential that a network meta-analysis might suffer from inconsistency. Suppose that a network meta-analysis compares a total of K treatments. When it contains two-arm studies only, Lu and Ades (2006) propose to measure the ICDF as

$$ICDF^{LA} = T - K + 1,$$

where T is the total number of treatment pairs that are directly compared in the treatment network. This measure is interpreted as the number of independent treatment loops. However, it may not be feasibly calculated when the network meta-analysis contains multi-arm studies. Multi-arm studies provide intrinsically consistent evidence and complicate the calculation of the ICDF.

Alternatively, the ICDF may be measured as the difference between the effective numbers of parameters of the consistency and inconsistency models for network meta-analysis. The consistency

model assumes evidence consistency in all treatment loops, and the inconsistency model treats the overall effect sizes of all direct treatment comparisons as separate, unrelated parameters (Dias et al., 2013). The effective number of parameters is frequently used to assess the complexity of Bayesian hierarchical models (Spiegelhalter et al., 2002). The effect measure is the (log) odds ratio in the models used in this function. Let $p_D^{FE,con}$ and $p_D^{FE,incon}$ be the effective numbers of parameters of the consistency and inconsistency models under the fixed-effects setting, and $p_D^{RE,con}$ and $p_D^{RE,incon}$ be those under the random-effects setting. The ICDF measures under the fixed-effects and random-effects settings are

$$ICDF^{FE} = p_D^{FE,incon} - p_D^{FE,con};$$

$$ICDF^{RE} = p_D^{RE,incon} - p_D^{RE,con},$$

respectively. See more details in Lin (2020).

Value

This function produces a list containing the following results: a table of the number of arms within studies and the corresponding counts of studies (`nstudy.trtarm`); the number of multi-arm studies (`nstudy.multi`); the set of treatments compared in each multi-arm study (`multi.trtarm`); the Lu–Ades ICDF measure (`icdf.la`); the ICDF measure based on the fixed-effects consistency and inconsistency models (`icdf.fe`); and the ICDF measure based on the random-effects consistency and inconsistency models (`icdf.re`). The Lu–Ades ICDF measure is `NA` (not available) in the presence of multi-arm studies, because multi-arm studies complicate the counting of independent treatment loops in generic network meta-analyses. When `traceplot = TRUE`, the trace plots will be saved in users' working directory.

References

Dias S, Welton NJ, Sutton AJ, Caldwell DM, Lu G, Ades AE (2013). "Evidence synthesis for decision making 4: inconsistency in networks of evidence based on randomized controlled trials." *Medical Decision Making*, **33**(5), 641–656. <doi:10.1177/0272989X12455847>

Lin L (2021). "Evidence inconsistency degrees of freedom in Bayesian network meta-analysis." *Journal of Biopharmaceutical Statistics*, **31**(3), 317–330. <doi:10.1080/10543406.2020.1852247>

Lu G, Ades AE (2006). "Assessing evidence inconsistency in mixed treatment comparisons." *Journal of the American Statistical Association*, **101**(474), 447–459. <doi:10.1198/016214505000001302>

Spiegelhalter DJ, Best NG, Carlin BP, Van Der Linde A (2002). "Bayesian measures of model complexity and fit." *Journal of the Royal Statistical Society, Series B (Statistical Methodology)*, **64**(4), 583–639. <doi:10.1111/14679868.00353>

Examples

```
data("dat.baker")
## increase n.burnin (e.g., to 50000) and n.iter (e.g., to 200000)
##  for better convergence of MCMC
out <- nma.icdf(sid, tid, r, n, data = dat.baker,
  type = c("la", "fe", "re"),
  n.adapt = 1000, n.chains = 3, n.burnin = 500, n.iter = 2000,
  n.thin = 2, traceplot = FALSE, seed = 1234)
```

```
out
```

`nma.predrank` *Predictive P-Score for Treatment Ranking in Bayesian Network Meta-Analysis*

Description

Calculates the P-score and predictive P-score for a network meta-analysis in the Bayesian framework described in Rosenberger et al. (2021).

Usage

```
nma.predrank(sid, tid, r, n, data, n.adapt = 1000, n.chains = 3, n.burnin = 2000,
             n.iter = 5000, n.thin = 2, lowerbetter = TRUE, pred = TRUE,
             pred.samples = FALSE, trace = FALSE)
```

Arguments

`sid`	a vector specifying the study IDs, from 1 to the number of studies.
`tid`	a vector specifying the treatment IDs, from 1 to the number of treatments.
`r`	a numeric vector specifying the event counts.
`n`	a numeric vector specifying the sample sizes.
`data`	an optional data frame containing the network meta-analysis dataset. If `data` is specified, the previous arguments, `sid`, `tid`, `r`, and `n`, should be specified as their corresponding column names in `data`.
`n.adapt`	the number of iterations for adaptation in the Markov chain Monte Carlo (MCMC) algorithm. The default is 1,000. This argument and the following `n.chains`, `n.burnin`, `n.iter`, and `n.thin` are passed to the functions in the package **rjags**.
`n.chains`	the number of MCMC chains. The default is 3.
`n.burnin`	the number of iterations for burn-in period. The default is 2,000.
`n.iter`	the total number of iterations in each MCMC chain after the burn-in period. The default is 5,000.
`n.thin`	a positive integer specifying thinning rate. The default is 2.
`lowerbetter`	A logical value indicating whether lower effect measures indicate better treaetments. If `lowerbetter` = `TRUE` (the default), then lower effect measures indicate better treatments.
`pred`	a logical value indicating whether the treatment ranking measures in a new study are to be derived. These measures are only derived when `pred` = `TRUE`.
`pred.samples`	a logical value indicating whether the posterior samples of expected scaled ranks in a new study are to be saved.
`trace`	a logical value indicating whether all posterior samples are to be saved.

Details

Under the frequentist setting, the P-score is built on the quantiles

$$P_{kh} = \Phi((\hat{d}_{1k} - \hat{d}_{1h})/s_{kh}),$$

where $\hat{d}_{1k}$ and $\hat{d}_{1h}$ are the point estimates of treatment effects for k vs. 1 and h vs. 1, respectively, and s_{kh} is the standard error of $\hat{d}_{1k} - \hat{d}_{1h}$ (Rucker and Schwarzer, 2015). Moreover, $\Phi(\cdot)$ is the cumulative distribution function of the standard normal distribution. The quantity P_{kh} can be interpreted as the extent of certainty that treatment k is better than h. The frequentist P-score of treatment k is $\frac{1}{K-1}\sum_{h\neq k} P_{kh}$.

Analogously to the frequentist P-score, conditional on d_{1h} and d_{1k}, the quantities P_{hk} from the Bayesian perspective can be considered as $I(d_{1k} > d_{1h})$, which are Bernoulli random variables. To quantify the overall performance of treatment k, we may similarly use

$$\bar{P}_k = \frac{1}{K-1}\sum_{h\neq k} I(d_{1k} > d_{1h}).$$

Note that $\bar{P}_k$ is a parameter under the Bayesian framework, while the frequentist P-score is a statistic. Moreover, $\sum_{h\neq k} I(d_{1k} > d_{1h})$ is equivalent to $K - R_k$, where R_k is the true rank of treatment k. Thus, we may also write $\bar{P}_k = (K-R_k)/(K-1)$; this corresponds to the findings by Rucker and Schwarzer (2015). Consequently, we call $\bar{P}_k$ the scaled rank in the network meta-analysis (NMA) for treatment k. It transforms the range of the original rank between 1 and K to a range between 0 and 1. In addition, note that $E[I(d_{1k} > d_{1h}|Data)] = \Pr(d_{1k} > d_{1h}|Data)$, which is analogous to the quantity of P_{kh} under the frequentist framework. Therefore, we use the posterior mean of the scaled rank $\bar{P}_k$ as the Bayesian P-score; it is a counterpart of the frequentist P-score.

The scaled ranks $\bar{P}_k$ can be feasibly estimated via the MCMC algorithm. Let $\{d_{1k}^{(j)}; k = 2, \ldots, K\}_{j=1}^{J}$ be the posterior samples of the overall relative effects d_{1k} of all treatments vs. the reference treatment 1 in a total of J MCMC iterations after the burn-in period, where j indexes the iterations. As d_{11} is trivially 0, we set $d_{11}^{(j)}$ to 0 for all j. The jth posterior sample of treatment k's scaled rank is $\bar{P}_k^{(j)} = \frac{1}{K-1}\sum_{h\neq k} I(d_{1k}^{(j)} > d_{1h}^{(j)})$. We can make inferences for the scaled ranks from the posterior samples $\{\bar{P}_k^{(j)}\}_{j=1}^{J}$, and use their posterior means as the Bayesian P-scores. We may also obtain the posterior medians as another set of point estimates, and the 2.5% and 97.5% posterior quantiles as the lower and upper bounds of 95% credible intervals (CrIs), respectively. Because the posterior samples of the scaled ranks take discrete values, the posterior medians and the CrI bounds are also discrete.

Based on the idea of the Bayesian P-score, we can similarly define the predictive P-score for a future study by accounting for the heterogeneity between the existing studies in the NMA and the new study. Specifically, we consider the probabilities in the new study

$$P_{new,kh} = \Pr(\delta_{new,1k} > \delta_{new,1h}),$$

conditional on the population parameters d_{1h}, d_{1k}, and τ from the NMA. Here, $\delta_{new,1k}$ and $\delta_{new,1h}$ represent the treatment effects of k vs. 1 and h vs. 1 in the new study, respectively. The $P_{new,kh}$ corresponds to the quantity P_{kh} in the NMA; it represents the probability of treatment k being better than h in the new study. Due to heterogeneity, $\delta_{new,1k} \sim N(d_{1k}, \tau^2)$ and $\delta_{new,1h} \sim N(d_{1h}, \tau^2)$. The correlation coefficients between treatment comparisons are typically assumed to be 0.5; therefore, such probabilities in the new study can be explicitly calculated as $P_{new,kh} = \Phi((d_{1k} -$

$d_{1h})/\tau)$, which is a function of d_{1h}, d_{1k}, and τ. Finally, we use

$$\bar{P}_{new,k} = \frac{1}{K-1}\sum_{h \neq k} P_{new,kh}$$

to quantify the performance of treatment k in the new study. The posterior samples of $\bar{P}_{new,k}$ can be derived from the posterior samples of d_{1k}, d_{1h}, and τ during the MCMC algorithm.

Note that the probabilities $P_{new,kh}$ can be written as $E[I(\delta_{new,1k} > \delta_{new,1h})]$. Based on similar observations for the scaled ranks in the NMA, the $\bar{P}_{new,k}$ in the new study subsequently becomes

$$\bar{P}_{new,k} = \frac{1}{K-1} E\left[\sum_{h \neq k} I(\delta_{new,1k} > \delta_{new,1h})\right] = E\left[\frac{K - R_{new,k}}{K-1}\right],$$

where $R_{new,k}$ is the true rank of treatment k in the new study. Thus, we call $\bar{P}_{new,k}$ the expected scaled rank in the new study. Like the Bayesian P-score, we define the predictive P-score as the posterior mean of $\bar{P}_{new,k}$. The posterior medians and 95% CrIs can also be obtained using the MCMC samples of $\bar{P}_{new,k}$. See more details in Rosenberger et al. (2021).

Value

This function estimates the P-score for all treatments in a Bayesian NMA, estimates the predictive P-score (if `pred = TRUE`), gives the posterior samples of expected scaled ranks in a new study (if `pred.samples = TRUE`), and outputs all MCMC posterior samples (if `trace = TRUE`).

Author(s)

Kristine J. Rosenberger, Lifeng Lin

References

Rosenberger KJ, Duan R, Chen Y, Lin L (2021). "Predictive P-score for treatment ranking in Bayesian network meta-analysis." *BMC Medical Research Methodology*, **21**, 213. <doi:10.1186/s12874021013975>

Rucker G, Schwarzer G (2015). "Ranking treatments in frequentist network meta-analysis works without resampling methods." *BMC Medical Research Methodology*, **15**, 58. <doi:10.1186/s12874-01500608>

Examples

```
## increase n.burnin (e.g., to 50000) and n.iter (e.g., to 200000)
## for better convergence of MCMC
data("dat.sc")
set.seed(1234)
out1 <- nma.predrank(sid, tid, r, n, data = dat.sc, n.burnin = 500, n.iter = 2000,
  lowerbetter = FALSE, pred.samples = TRUE)
out1$P.score
out1$P.score.pred

cols <- c("red4", "plum4", "paleturquoise4", "palegreen4")
```

```
cols.hist <- adjustcolor(cols, alpha.f = 0.4)
trtnames <- c("1) No contact", "2) Self-help", "3) Individual counseling",
  "4) Group counseling")
brks <- seq(0, 1, 0.01)
hist(out1$P.pred[[1]], breaks = brks, freq = FALSE,
  xlim = c(0, 1), ylim = c(0, 5), col = cols.hist[1], border = cols[1],
  xlab = "Expected scaled rank in a new study", ylab = "Density", main = "")
hist(out1$P.pred[[2]], breaks = brks, freq = FALSE,
  col = cols.hist[2], border = cols[2], add = TRUE)
hist(out1$P.pred[[3]], breaks = brks, freq = FALSE,
  col = cols.hist[3], border = cols[3], add = TRUE)
hist(out1$P.pred[[4]], breaks = brks, freq = FALSE,
  col = cols.hist[4], border = cols[4], add = TRUE)
legend("topright", fill = cols.hist, border = cols, legend = trtnames)


data("dat.xu")
set.seed(1234)
out2 <- nma.predrank(sid, tid, r, n, data = dat.xu, n.burnin = 500, n.iter = 2000)
out2
```

`pb.bayesian.binary` *Bayesian Method for Assessing Publication Bias/Small-Study Effects in Meta-Analysis of a Binary Outcome*

Description

Performs multiple methods introduced in Shi et al. (2020) to assess publication bias/small-study effects under the Bayesian framework in a meta-analysis of (log) odds ratios.

Usage

```
pb.bayesian.binary(n00, n01, n10, n11, p01 = NULL, p11 = NULL, data,
                   sig.level = 0.1,  method = "bay", het = "mul",
                   sd.prior = "unif", n.adapt = 1000, n.chains = 3,
                   n.burnin = 5000, n.iter = 10000, thin = 2,
                   upp.het = 2, phi = 0.5, coda = FALSE,
                   traceplot = FALSE, seed = 1234)
```

Arguments

`n00`	a numeric vector or the corresponding column name in the argument `data`, specifying the counts of non-events in treatment group 0 in the collected studies.
`n01`	a numeric vector or the corresponding column name in the argument `data`, specifying the counts of events in treatment group 0 in the collected studies.
`n10`	a numeric vector or the corresponding column name in the argument `data`, specifying the counts of non-events in treatment group 1 in the collected studies.

n11	a numeric vector or the corresponding column name in the argument data, specifying the counts of events in treatment group 1 in the collected studies.
p01	an optional numeric vector specifying true event rates (e.g., from simulations) in the treatment group 0 across studies.
p11	an optional numeric vector specifying true event rates (e.g., from simulations) in the treatment group 1 across studies.
data	an optional data frame containing the meta-analysis dataset. If data is specified, the previous arguments, n00, n01, n10, n11, p01 (if any), and p11 (if any) should be specified as their corresponding column names in data.
sig.level	a numeric value specifying the statistical significance level α for testing for publication bias. The default is 0.1. It corresponds to $(1 - \alpha) \times 100\%$ confidence/credible intervals.
method	a character string specifying the method for assessing publication bias via Bayesian hierarchical models. It can be one of "bay" (the Bayesian approach proposed in Shi et al., 2020), "reg.bay" (Egger's regression test, see Egger et al., 1997, under the Bayesian framework) and "smoothed.bay" (the regression test based on the smoothed sample variance, see Jin et al., 2014, under the Bayesian framework), where all regression tests are under the random-effects setting. The default is "bay".
het	a character string specifying the type of heterogeneity assumption for the publication bias tests. It can be either "mul" (multiplicative heterogeneity assumption; see Thompson and Sharpe, 1999) or "add" (additive heterogeneity assumption). The default is "mul".
sd.prior	a character string specifying prior distributions for standard deviation parameters. It can be either "unif" (uniform distribution) or "hn" (half-normal distribution). The default is "unif".
n.adapt	the number of iterations for adaptation in the Markov chain Monte Carlo (MCMC) algorithm. The default is 1,000. This argument and the following n.chains, n.burnin, n.iter, and thin are passed to the functions in the package **rjags**.
n.chains	the number of MCMC chains. The default is 1.
n.burnin	the number of iterations for burn-in period. The default is 5,000.
n.iter	the total number of iterations in each MCMC chain after the burn-in period. The default is 10,000.
thin	a positive integer specifying thinning rate. The default is 2.
upp.het	a positive number for specifying the upper bound of uniform priors for standard deviation parameters (if sd.prior = "unif"). The default is 2.
phi	a positive number for specifying the hyper-parameter of half-normal priors for standard deviation parameters (if sd.prior = "hn"). The default is 0.5.
coda	a logical value indicating whether to output MCMC posterior samples. The default is FALSE.
traceplot	a logical value indicating whether to draw trace plots for the regression slopes. The default is FALSE.
seed	an integer for specifying the seed value for reproducibility.

Details

The Bayesian models are specified in Shi et al. (2020). The vague prior N(0, 10^4) is used for the regression intercept and slope, and the uniform prior U(0, `upp.het`) and half-normal prior HN(`phi`) are used for standard deviation parameters. The half-normal priors may be preferred in meta-analyses with rare events or small sample sizes.

Value

This function returns a list containing estimates of regression slopes and their credible intervals with the specified significance level (`sig.level`) as well as MCMC posterior samples (if `coda = TRUE`). Each element name in this list is related to a certain publication bias method (e.g., `est.bay` and `ci.bay` represent the slope estimate and its credible interval based on the proposed Bayesian method). In addition, trace plots for the regression slope are drawn if `traceplot = TRUE`.

Note

The current version does not support other effect measures such as relative risks or risk differences.

Author(s)

Linyu Shi <ls16d@my.fsu.edu>

References

Egger M, Davey Smith G, Schneider M, Minder C (1997). "Bias in meta-analysis detected by a simple, graphical test." *BMJ*, **315**(7109), 629–634. <doi:10.1136/bmj.315.7109.629>

Jin Z-C, Wu C, Zhou X-H, He J (2014). "A modified regression method to test publication bias in meta-analyses with binary outcomes." *BMC Medical Research Methodology*, **14**, 132. <doi:10.1186/1471228814132>

Shi L, Chu H, Lin L (2020). "A Bayesian approach to assessing small-study effects in meta-analysis of a binary outcome with controlled false positive rate". *Research Synthesis Methods*, **11**(4), 535–552. <doi:10.1002/jrsm.1415>

Thompson SG, Sharp SJ (1999). "Explaining heterogeneity in meta-analysis: a comparison of methods." *Statistics in Medicine*, **18**(20), 2693–2708. <doi:10.1002/(SICI)10970258(19991030)18:20<2693::AID-SIM235>3.0.CO;2V>

See Also

`pb.hybrid.binary`, `pb.hybrid.generic`

Examples

```
data("dat.poole")
set.seed(654321)
## increase n.burnin and n.iter for better convergence of MCMC
rslt.poole <- pb.bayesian.binary(n00, n01, n10, n11, data = dat.poole,
  method = "bay", het = "mul", sd.prior = "unif", n.adapt = 1000,
  n.chains = 3, n.burnin = 500, n.iter = 2000, thin = 2, upp.het = 2)
rslt.poole
```

```
data("dat.ducharme")
set.seed(654321)
## increase n.burnin and n.iter for better convergence of MCMC
rslt.ducharme <- pb.bayesian.binary(n00, n01, n10, n11, data = dat.ducharme,
  method = "bay", het = "mul", sd.prior = "unif", n.adapt = 1000,
  n.chains = 3, n.burnin = 500, n.iter = 2000, thin = 2, upp.het = 2)
rslt.ducharme

data("dat.henry")
set.seed(654321)
## increase n.burnin and n.iter for better convergence of MCMC
rslt.henry <- pb.bayesian.binary(n00, n01, n10, n11, data = dat.henry,
  method = "bay", het = "mul", sd.prior = "unif", n.adapt = 1000,
  n.chains = 3, n.burnin = 500, n.iter = 2000, thin = 2, upp.het = 2)
rslt.henry
```

`pb.hybrid.binary` *Hybrid Test for Publication Bias/Small-Study Effects in Meta-Analysis With Binary Outcomes*

Description

Performs the hybrid test for publication bias/small-study effects introduced in Lin (2020), which synthesizes results from multiple popular publication bias tests, in a meta-analysis with binary outcomes.

Usage

```
pb.hybrid.binary(n00, n01, n10, n11, data, methods,
                 iter.resam = 1000, theo.pval = TRUE)
```

Arguments

`n00`	a numeric vector or the corresponding column name in the argument `data`, specifying the counts of non-events in treatment group 0 in the collected studies.
`n01`	a numeric vector or the corresponding column name in the argument `data`, specifying the counts of events in treatment group 0 in the collected studies.
`n10`	a numeric vector or the corresponding column name in the argument `data`, specifying the counts of non-events in treatment group 1 in the collected studies.
`n11`	a numeric vector or the corresponding column name in the argument `data`, specifying the counts of events in treatment group 1 in the collected studies.
`data`	an optional data frame containing the meta-analysis dataset. If `data` is specified, the previous arguments, `n00`, `n01`, `n10`, and `n11`, should be specified as their corresponding column names in `data`.

`methods` a vector of character strings specifying the publication bias tests to be included in the hybrid test. They can be a subset of `"rank"` (Begg's rank test; see Begg and Mazumdar, 1994), `"reg"` (Egger's regression test under the fixed-effect setting; see Egger et al., 1997), `"reg.het"` (Egger's regression test accounting for additive heterogeneity), `"skew"` (the skewness-based test under the fixed-effect setting; see Lin and Chu, 2018), `"skew.het"` (the skewness-based test accounting for additive heterogeneity), `"inv.sqrt.n"` (the regression test based on sample sizes; see Tang and Liu, 2000), `"trimfill"` (the trim-and-fill method; see Duval and Tweedie, 2000), `"n"` (the regressoin test with sample sizes as the predictor; see Macaskill et al., 2001), `"inv.n"` (the regressoin test with the inverse of sample sizes as the predictor; see Peters et al., 2006), `"as.rank"` (the rank test based on the arcsine-transformed effect sizes; see Rucker et al., 2008), `"as.reg"` (the regression test based on the arcsine-transformed effect sizes under the fixed-effect setting), `"as.reg.het"` (the regression test based on the arcsine-transformed effect sizes accounting for additive heterogeneity), `"smoothed"` (the regression test based on the smoothed sample variances under the fixed-effect setting; see Jin et al., 2014), `"smoothed.het"` (the regression test based on the smoothed sample variances accounting for additive heterogeneity), `"score"` (the regression test based on the score function; see Harbord et al., 2006), and `"count"` (the test based on the hypergeometric distributions of event counts, designed for sparse data; see Schwarzer et al., 2007). The default is to include all aforementioned tests.

`iter.resam` a positive integer specifying the number of resampling iterations for calculating the p-value of the hybrid test.

`theo.pval` a logical value indicating whether additionally calculating the p-values of the tests specified in `methods` based on the test statistics' theoretical null distributions. Regardless of this argument, the resampling-based p-values are always produced by this function for the tests specified in `methods`.

Details

The hybrid test statistic is defined as the minimum p-value among the publication bias tests considered in the set specified by the argument `methods`. Note that the minimum p-value is no longer a genuine p-value because it cannot control the type I error rate. Its p-value needs to be calculated via the resampling approach. See more details in Lin (2020).

Value

This function returns a list containing p-values of the publication bias tests specified in `methods` as well as the hybrid test. Each element's name in this list has the format of `pval.x`, where `x` stands for the character string corresponding to a certain publication bias test, such as `rank`, `reg`, `skew`, etc. The hybrid test's p-value has the name `pval.hybrid`. If `theo.pval = TRUE`, additional elements of p-values of the tests in `methods` based on theorectical null distributions are included in the produced list; their names have the format of `pval.x.theo`. Another p-value of the hybrid test is also produced based on them; its corresponding element has the name `pval.hybrid.theo`.

References

Begg CB, Mazumdar M (1994). "Operating characteristics of a rank correlation test for publication bias." *Biometrics*, **50**(4), 1088–1101. <doi:10.2307/2533446>

Duval S, Tweedie R (2000). "A nonparametric 'trim and fill' method of accounting for publication bias in meta-analysis." *Journal of the American Statistical Association*, **95**(449), 89–98. <doi:10.1080/01621459.2000.10473905>

Egger M, Davey Smith G, Schneider M, Minder C (1997). "Bias in meta-analysis detected by a simple, graphical test." *BMJ*, **315**(7109), 629–634. <doi:10.1136/bmj.315.7109.629>

Harbord RM, Egger M, Sterne JAC (2006). "A modified test for small-study effects in meta-analyses of controlled trials with binary endpoints." *Statistics in Medicine*, **25**(20), 3443–3457. <doi:10.1002/sim.2380>

Jin Z-C, Wu C, Zhou X-H, He J (2014). "A modified regression method to test publication bias in meta-analyses with binary outcomes." *BMC Medical Research Methodology*, **14**, 132. <doi:10.1186/1471228814132>

Lin L (2020). "Hybrid test for publication bias in meta-analysis." *Statistical Methods in Medical Research*, **29**(10), 2881–2899. <doi:10.1177/0962280220910172>

Lin L, Chu H (2018). "Quantifying publication bias in meta-analysis." *Biometrics*, **74**(3), 785–794. <doi:10.1111/biom.12817>

Macaskill P, Walter SD, Irwig L (2001). "A comparison of methods to detect publication bias in meta-analysis." *Statistics in Medicine*, **20**(4), 641–654. <doi:10.1002/sim.698>

Peters JL, Sutton AJ, Jones DR, Abrams KR, Rushton L (2006). "Comparison of two methods to detect publication bias in meta-analysis." *JAMA*, **295**(6), 676–680. <doi:10.1001/jama.295.6.676>

Rucker G, Schwarzer G, Carpenter J (2008). "Arcsine test for publication bias in meta-analyses with binary outcomes." *Statistics in Medicine*, **27**(5), 746–763. <doi:10.1002/sim.2971>

Schwarzer G, Antes G, Schumacher M (2007). "A test for publication bias in meta-analysis with sparse binary data." *Statistics in Medicine*, **26**(4), 721–733. <doi:10.1002/sim.2588>

Tang J-L, Liu JLY (2000). "Misleading funnel plot for detection of bias in meta-analysis." *Journal of Clinical Epidemiology*, **53**(5), 477–484. <doi:10.1016/S08954356(99)002048>

Thompson SG, Sharp SJ (1999). "Explaining heterogeneity in meta-analysis: a comparison of methods." *Statistics in Medicine*, **18**(20), 2693–2708. <doi:10.1002/(SICI)10970258(19991030)18:20<2693::AIDSIM235>3.0.CO;2V>

See Also

`pb.bayesian.binary`, `pb.hybrid.generic`

Examples

```
## meta-analysis of (log) odds ratios
data("dat.whiting")
# based on only 10 resampling iterations
set.seed(1234)
out.whiting <- pb.hybrid.binary(n00 = n00, n01 = n01,
  n10 = n10, n11 = n11, data = dat.whiting, iter.resam = 10)
out.whiting
```

```
# increases the number of resampling iterations to 10000,
#  taking longer time
```

`pb.hybrid.generic` *Hybrid Test for Publication Bias/Small-Study Effects in Meta-Analysis With Generic Outcomes*

Description

Performs the hybrid test for publication bias/small-study effects introduced in Lin (2020), which synthesizes results from multiple popular publication bias tests, in a meta-analysis with generic outcomes.

Usage

```
pb.hybrid.generic(y, s2, n, data, methods,
                  iter.resam = 1000, theo.pval = TRUE)
```

Arguments

`y`	a numeric vector or the corresponding column name in the argument `data`, specifying the observed effect sizes in the collected studies.
`s2`	a numeric vector or the corresponding column name in the argument `data`, specifying the within-study variances.
`n`	an optional numeric vector or the corresponding column name in the argument `data`, specifying the study-specific total sample sizes. This argument is required if the sample-size-based test (`"inv.sqrt.n"`) is included in `method`.
`data`	an optional data frame containing the meta-analysis dataset. If `data` is specified, the previous arguments, `y`, `s2`, and `n`, should be specified as their corresponding column names in `data`.
`methods`	a vector of character strings specifying the publication bias tests to be included in the hybrid test. They can be a subset of `"rank"` (Begg's rank test; see Begg and Mazumdar, 1994), `"reg"` (Egger's regression test under the fixed-effect setting; see Egger et al., 1997), `"reg.het"` (Egger's regression test accounting for additive heterogeneity), `"skew"` (the skewness-based test under the fixed-effect setting; see Lin and Chu, 2018), `"skew.het"` (the skewness-based test accounting for additive heterogeneity), `"inv.sqrt.n"` (the regression test based on sample sizes; see Tang and Liu, 2000), and `"trimfill"` (the trim-and-fill method; see Duval and Tweedie, 2000). The default is to include all aforementioned tests.
`iter.resam`	a positive integer specifying the number of resampling iterations for calculating the p-value of the hybrid test.
`theo.pval`	a logical value indicating whether additionally calculating the p-values of the tests specified in `methods` based on the test statistics' theoretical null distributions. Regardless of this argument, the resampling-based p-values are always produced by this function for the tests specified in `methods`.

Details

The hybrid test statistic is defined as the minimum p-value among the publication bias tests considered in the set specified by the argument `methods`. Note that the minimum p-value is no longer a genuine p-value because it cannot control the type I error rate. Its p-value needs to be calculated via the resampling approach. See more details in Lin (2020).

Value

This function returns a list containing p-values of the publication bias tests specified in `methods` as well as the hybrid test. Each element's name in this list has the format of `pval.x`, where `x` stands for the character string corresponding to a certain publication bias test, such as `rank`, `reg`, `skew`, etc. The hybrid test's p-value has the name `pval.hybrid`. If `theo.pval = TRUE`, additional elements of p-values of the tests in `methods` based on theorectical null distributions are included in the produced list; their names have the format of `pval.x.theo`. Another p-value of the hybrid test is also produced based on them; its corresponding element has the name `pval.hybrid.theo`.

References

Begg CB, Mazumdar M (1994). "Operating characteristics of a rank correlation test for publication bias." *Biometrics*, **50**(4), 1088–1101. <doi:10.2307/2533446>

Duval S, Tweedie R (2000). "A nonparametric 'trim and fill' method of accounting for publication bias in meta-analysis." *Journal of the American Statistical Association*, **95**(449), 89–98. <doi:10.1080/01621459.2000.10473905>

Egger M, Davey Smith G, Schneider M, Minder C (1997). "Bias in meta-analysis detected by a simple, graphical test." *BMJ*, **315**(7109), 629–634. <doi:10.1136/bmj.315.7109.629>

Lin L (2020). "Hybrid test for publication bias in meta-analysis." *Statistical Methods in Medical Research*, **29**(10), 2881–2899. <doi:10.1177/0962280220910172>

Lin L, Chu H (2018). "Quantifying publication bias in meta-analysis." *Biometrics*, **74**(3), 785–794. <doi:10.1111/biom.12817>

Tang J-L, Liu JLY (2000). "Misleading funnel plot for detection of bias in meta-analysis." *Journal of Clinical Epidemiology*, **53**(5), 477–484. <doi:10.1016/S08954356(99)002048>

Thompson SG, Sharp SJ (1999). "Explaining heterogeneity in meta-analysis: a comparison of methods." *Statistics in Medicine*, **18**(20), 2693–2708. <doi:10.1002/(SICI)10970258(19991030)18:20<2693::AIDSIM235>3.0.CO;2V>

See Also

`pb.bayesian.binary`, `pb.hybrid.binary`

Examples

```
## meta-analysis of mean differences
data("dat.plourde")
# based on only 10 resampling iterations
set.seed(1234)
out.plourde <- pb.hybrid.generic(y = y, s2 = s2, n = n,
  data = dat.plourde, iter.resam = 10)
```

```
out.plourde
# only produces resampling-based p-values
set.seed(1234)
pb.hybrid.generic(y = y, s2 = s2, n = n,
  data = dat.plourde, iter.resam = 10, theo.pval = FALSE)
# increases the number of resampling iterations to 10000,
#  taking longer time

## meta-analysis of standardized mean differences
data("dat.paige")
# based on only 10 resampling iterations
set.seed(1234)
out.paige <- pb.hybrid.generic(y = y, s2 = s2, n = n,
  data = dat.paige, iter.resam = 10)
out.paige
# increases the number of resampling iterations to 10000,
#  taking longer time
```

`plot.meta.dt` *Plot for Meta-Analysis of Diagnostic Tests*

Description

Visualizes meta-analysis of diagnostic tests by presenting summary results, such as ROC (receiver operating characteristic) curve, overall sensitivity and overall specificity (1 − specificity), and their confidence and prediction regions.

Usage

```
## S3 method for class 'meta.dt'
plot(x, add = FALSE, xlab, ylab, alpha,
     studies = TRUE, cex.studies, col.studies, pch.studies,
     roc, col.roc, lty.roc, lwd.roc, weight = FALSE,
     eqline, col.eqline, lty.eqline, lwd.eqline,
     overall = TRUE, cex.overall, col.overall, pch.overall,
     confid = TRUE, col.confid, lty.confid, lwd.confid,
     predict = FALSE, col.predict, lty.predict, lwd.predict, ...)
```

Arguments

`x`	an object of class `"meta.dt"` created by the function `meta.dt`.
`add`	a logical value indicating if the plot is added to an already existing plot.
`xlab`	a label for the x axis; the default is "1 - Specificity".
`ylab`	a label for the y axis; the default is "Sensitivity".
`alpha`	a numeric value specifying the statistical significance level for the confidence and prediction regions. If not specified, the plot uses the significance level stored in `x` (i.e., `x$alpha`).

studies	a logical value indicating if the individual studies are presented in the plot.
cex.studies	the size of points representing individual studies (the default is 1).
col.studies	the color of points representing individual studies (the default is "black").
pch.studies	the symbol of points representing individual studies (the default is 1, i.e., circle).
roc	a logical value indicating if the ROC curve is presented in the plot. The default is TRUE for the summary ROC approach (x$method = "s.roc") and is FALSE for the bivariate (generalized) linear mixed model (x$method = "biv.lmm" or "biv.glmm").
col.roc	the color of the ROC curve (the default is "black").
lty.roc	the line type of the ROC curve (the default is 1, i.e., solid line).
lwd.roc	the line width of the ROC curve (the default is 1).
weight	a logical value indicating if the weighted (TRUE) or unweighted (FALSE, the default) regression is used for the summary ROC approach (when x$method is "s.roc").
eqline	a logical value indicating if the line of sensitivity equaling to specificity is presented in the plot.
col.eqline	the color of the equality line (the default is "black").
lty.eqline	the type of the equality line (the default is 4, i.e., dot-dash line).
lwd.eqline	the width of the equality line (the default is 1).
overall	a logical value indicating if the overall sensitivity and overall specificity are presented in the plot. This and the following arguments are used for the bivariate (generalized) linear mixed model (x$method = "biv.lmm" or "biv.glmm").
cex.overall	the size of the point representing the overall sensitivity and overall specificity (the default is 1).
col.overall	the color of the point representing the overall sensitivity and overall specificity (the default is "black").
pch.overall	the symbol of the point representing the overall sensitivity and overall specificity (the default is 15, i.e., filled square).
confid	a logical value indicating if the confidence region of the overall sensitivity and overall specificity is presented in the plot.
col.confid	the line color of the confidence region (the default is "black").
lty.confid	the line type of the confidence region (the default is 2, i.e., dashed line).
lwd.confid	the line width of the confidence region (the default is 1).
predict	a logical value indicating if the prediction region of the overall sensitivity and overall specificity is presented in the plot.
col.predict	the line color of the prediction region (the default is "black").
lty.predict	the line type of the prediction region (the default is 3, i.e., dotted line).
lwd.predict	the line width of the prediction region (the default is 1).
...	other arguments that can be passed to the function plot.default.

Value

None.

See Also

meta.dt, print.meta.dt

---

plot.metaoutliers *Standardized Residual Plot for Outliers Diagnostics*

Description

Draws a plot showing study-specific standardized residuals.

Usage

```
## S3 method for class 'metaoutliers'
plot(x, xtick.cex = 1, ytick.cex = 0.5, ...)
```

Arguments

x	an object created by the function metaoutliers.
xtick.cex	a numeric value specifying the size of ticks on the x axis.
ytick.cex	a numeric value specifying the size of ticks on the y axis.
...	other arguments that can be passed to the function plot.default.

Value

None.

See Also

metaoutliers

Examples

```
data("dat.aex")
attach(dat.aex)
out.aex <- metaoutliers(y, s2, model = "FE")
detach(dat.aex)
plot(out.aex)

data("dat.hipfrac")
attach(dat.hipfrac)
out.hipfrac <- metaoutliers(y, s2, model = "RE")
detach(dat.hipfrac)
plot(out.hipfrac)
```

print.meta.dt *Print Method for "meta.dt" Objects*

Description

Prints information about a meta-analysis of diagnostic tests.

Usage

```
## S3 method for class 'meta.dt'
print(x, digits = 3, ...)
```

Arguments

x	an object of class "meta.dt" produced by the function meta.dt.
digits	an integer specifying the number of decimal places to which the printed results should be rounded.
...	other arguments.

Value

None.

See Also

meta.dt, plot.meta.dt

ssfunnel *Contour-Enhanced Sample-Size-Based Funnel Plot*

Description

Generates contour-enhanced sample-size-based funnel plot for a meta-analysis of mean differences, standardized mean differences, (log) odds ratios, (log) relative risks, or risk differences.

Usage

```
ssfunnel(y, s2, n, data, type, alpha = c(0.1, 0.05, 0.01, 0.001),
         log.ss = FALSE, sigma, p0, xlim, ylim, xlab, ylab,
         cols.contour, col.mostsig, cex.pts, lwd.contour, pch,
         x.legend, y.legend, cex.legend, bg.legend, ...)
```

Arguments

y	a numeric vector or the corresponding column name in the argument data, specifying the observed effect sizes in the collected studies.
s2	a numeric vector or the corresponding column name in the argument data, specifying the within-study variances.
n	a numeric vector or the corresponding column name in the argument data, specifying the study-specific total sample sizes.
data	an optional data frame containing the meta-analysis dataset. If data is specified, the previous arguments, y, s2, and n, should be specified as their corresponding column names in data.
type	a character string specifying the type of effect size, which should be one of "md" (mean difference), "smd" (standardized mean difference), "lor" (log odds ratio), "lrr" (log relative risk), and "rd" (risk difference).
alpha	a numeric vector specifying the significance levels to be presented in the sample-size-based funnel plot.
log.ss	a logical value indicating whether sample sizes are plotted on a logarithmic scale (TRUE) or not (FALSE, the default).
sigma	a positive numeric value that is required for the mean difference (type = "md"), specifying a rough estimate of the common standard deviation of the samples' continuous outcomes in the two groups across studies. It is not used for other effect size types.
p0	an optional numeric value specifying a rough estimate of the common event rate in the control group across studies. It is only used for the (log) odds ratio, (log) relative risk, and risk difference.
xlim	the x limits c(x1, x2) of the plot.
ylim	the y limits c(y1, y2) of the plot.
xlab	a label for the x axis.
ylab	a label for the y axis.
cols.contour	a vector of character strings; they indicate colors of the contours to be presented in the sample-size-based funnel plot, and correspond to the significance levels specified in the argument alpha.
col.mostsig	a character string specifying the color for the most significant result among the studies in the meta-analysis.
cex.pts	the size of the points.
lwd.contour	the width of the contours.
pch	the symbol of the points.
x.legend	the x co-ordinate or a keyword, such as "topleft" (the default), to be used to position the legend. It is passed to legend.
y.legend	the y co-ordinate to be used to position the legend (the default is NULL).
cex.legend	the size of legend text.
bg.legend	the background color for the legend box.
...	other arguments that can be passed to plot.default.

Details

A contour-enhanced sample-size-based funnel plot is generated; it presents study-specific total sample sizes against the corresponding effect size estimates. It is helpful to avoid the confounding effect caused by the intrinsic association between effect size estimates and standard errors in the conventional standard-error-based funnel plot. See details of the derivations of the contours in Lin (2019).

Value

None.

References

Lin L (2019). "Graphical augmentations to sample-size-based funnel plot in meta-analysis." *Research Synthesis Methods*, **10**(3), 376–388. <doi:10.1002/jrsm.1340>

Peters JL, Sutton AJ, Jones DR, Abrams KR, Rushton L (2006). "Comparison of two methods to detect publication bias in meta-analysis." *JAMA*, **295**(6), 676–680. <doi:10.1001/jama.295.6.676>

Examples

```
## mean difference
data("dat.annane")
# descriptive statistics for sigma (continuous outcomes' standard deviation)
quantile(sqrt(dat.annane$s2/(1/dat.annane$n1 + 1/dat.annane$n2)),
  probs = c(0, 0.25, 0.5, 0.75, 1))
# based on sigma = 8
ssfunnel(y, s2, n, data = dat.annane, type = "md",
  alpha = c(0.1, 0.05, 0.01, 0.001), sigma = 8)
# sample sizes presented on a logarithmic scale with plot title
ssfunnel(y, s2, n, data = dat.annane, type = "md",
  alpha = c(0.1, 0.05, 0.01, 0.001), sigma = 8, log.ss = TRUE,
  main = "Contour-enhanced sample-size-based funnel plot")
# based on sigma = 17, with specified x and y limits
ssfunnel(y, s2, n, data = dat.annane, type = "md",
  xlim = c(-15, 15), ylim = c(30, 500),
  alpha = c(0.1, 0.05, 0.01, 0.001), sigma = 17, log.ss = TRUE)
# based on sigma = 20
ssfunnel(y, s2, n, data = dat.annane, type = "md",
  xlim = c(-15, 15), ylim = c(30, 500),
  alpha = c(0.1, 0.05, 0.01, 0.001), sigma = 20, log.ss = TRUE)

## standardized mean difference
data("dat.barlow")
ssfunnel(y, s2, n, data = dat.barlow, type = "smd",
  alpha = c(0.1, 0.05, 0.01, 0.001), xlim = c(-1.5, 1))

## log odds ratio
data("dat.butters")
ssfunnel(y, s2, n, data = dat.butters, type = "lor",
  alpha = c(0.1, 0.05, 0.01, 0.001), xlim = c(-1.5, 1.5))
# use different colors for contours
ssfunnel(y, s2, n, data = dat.butters, type = "lor",
```

```
  alpha = c(0.1, 0.05, 0.01, 0.001), xlim = c(-1.5, 1.5),
  cols.contour = c("blue", "green", "yellow", "red"), col.mostsig = "black")
# based on p0 = 0.3 (common event rate in the control group across studies)
ssfunnel(y, s2, n, data = dat.butters, type = "lor",
  alpha = c(0.1, 0.05, 0.01, 0.001), xlim = c(-1.5, 1.5), p0 = 0.3)
# based on p0 = 0.5
ssfunnel(y, s2, n, data = dat.butters, type = "lor",
  alpha = c(0.1, 0.05, 0.01, 0.001), xlim = c(-1.5, 1.5), p0 = 0.5)

## log relative risk
data("dat.williams")
ssfunnel(y, s2, n, data = dat.williams, type = "lrr",
  alpha = c(0.1, 0.05, 0.01, 0.001), xlim = c(-1.5, 2.5))
# based on p0 = 0.2
ssfunnel(y, s2, n, data = dat.williams, type = "lrr",
  alpha = c(0.1, 0.05, 0.01, 0.001), p0 = 0.2, xlim = c(-1.5, 2.5))
# based on p0 = 0.3
ssfunnel(y, s2, n, data = dat.williams, type = "lrr",
  alpha = c(0.1, 0.05, 0.01, 0.001), p0 = 0.3, xlim = c(-1.5, 2.5))

## risk difference
data("dat.kaner")
ssfunnel(y, s2, n, data = dat.kaner, type = "rd",
  alpha = c(0.1, 0.05, 0.01, 0.001), xlim = c(-0.5, 0.5))
# based on p0 = 0.1
ssfunnel(y, s2, n, data = dat.kaner, type = "rd",
  alpha = c(0.1, 0.05, 0.01, 0.001), p0 = 0.1, xlim = c(-0.5, 0.5))
# based on p0 = 0.4
ssfunnel(y, s2, n, data = dat.kaner, type = "rd",
  alpha = c(0.1, 0.05, 0.01, 0.001), p0 = 0.4, xlim = c(-0.5, 0.5))
```

Index