Meta-analysis based on Sisks et al. (2018) - V.1

Giulia Bertoldo

4/7/2022

Contents


```r
# Load packages
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(meta)
```

```
## Warning: package 'meta' was built under R version 4.0.5

## Loading 'meta' package (version 5.2-0).
## Type 'help(meta)' for a brief overview.
```

```
## Readers of 'Meta-Analysis with R (Use R!)' should install
## older version of 'meta' package: https://tinyurl.com/dt4y5drs
```

```r
library(readxl)

# Import data
df1 <- read_excel('data/mindset.xlsx', sheet = 'Meta-analysis 1')
```

Data cleaning

```r
# Glimpse data
glimpse(df1)
```

```
## Rows: 273
## Columns: 35
## $ `Document #`                          <dbl> 1, 2, 2, 2, 3, 3, 3, 3, 4, 5, 6...
## $ `Study #`                             <dbl> 1, 2, 2, 2, 3, 3, 3, 3, 4, 5, 6...
## $ `Sample #`                            <dbl> 1, 2, 3, 4, 5, 163, 164, 165, 6...
## $ `Sample Country`                      <chr> "Indonesia", "USA", "USA", "USA...
## $ `ES #`                                <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, ...
## $ Reference                             <chr> "Adatitomo (2015)", "Bagley (20...
## $ N                                     <dbl> 123, 400, 1019, 710, 250, 272, ...
## $ `Adjusted N`                          <dbl> 123.000000, 400.000000, 1019.00...
## $ `Student Description`                 <chr> "second semester university stu...
## $ `School Level`                        <chr> "post-secondary", "post-seconda...
## $ `Development Stage`                   <chr> "Adults", "Adults", "Adults", "...
## $ `Risk status`                         <chr> "low", "moderate", "moderate", ...
## $ SES                                   <chr> "not reported", "not reported",...
## $ `MS Measure`                          <chr> "Mindset about intelligence", "...
## $ `MS Measure Description`              <chr> "6 items, 3 growth and 3 fixed ...
## $ `Mindset Type`                        <chr> "Intelligence", "Personal attri...
## $ `Achievement Measure Description`     <chr> "Statistics final exam grade", ...
## $ `Academic Achievement Measure Type`   <chr> "Course exam", "Course grade", ...
## $ `Lab-based`                           <chr> "no", "no", "no", "no", "no", "...
## $ Published                             <chr> "yes", "no", "no", "no", "no", ...
## $ `ES type`                             <chr> "continuous", "continuous", "co...
## $ Calculation                           <chr> "Pearson's r", "sqrt of bivaria...
## $ Variance                              <dbl> 0.0079425749, 0.0024188215, 0.0...
## $ `Adjusted Variance`                   <dbl> 0.0079425749, 0.0024188215, 0.0...
## $ `Significant?`                        <chr> "N", "Y", "Y", "Y", "Y", "Y", "...
## $ r                                     <dbl> -0.12500000, 0.13266499, 0.1972...
## $ `Growth M`                            <dbl> NA, NA, NA, NA, NA, NA, NA, NA,...
## $ `Growth SD`                           <dbl> NA, NA, NA, NA, NA, NA, NA, NA,...
## $ `Other M`                             <dbl> NA, NA, NA, NA, NA, NA, NA, NA,...
## $ `Other SD`                            <dbl> NA, NA, NA, NA, NA, NA, NA, NA,...
## $ `Cohen's d`                           <dbl> NA, NA, NA, NA, NA, NA, NA, NA,...
## $ rpb                                   <dbl> NA, NA, NA, NA, NA, NA, NA, NA,...
## $ rb                                    <dbl> NA, NA, NA, NA, NA, NA, NA, NA,...
## $ `Calculated r`                        <dbl> NA, NA, NA, NA, NA, NA, NA, NA,...
## $ Notes                                 <chr> "the authors of the study also ...
```

```r
# Rename columns
df2 <- rename(df1,
             document_id = 'Document #',
             study_id = 'Study #',
             sample_id = 'Sample #',
             sample_country = 'Sample Country',
             es_id = 'ES #',
             reference = 'Reference',
             n = N,
             adjusted_n = 'Adjusted N',
             student_description = 'Student Description',
```

```r
             school_level = 'School Level',
             development_stage = 'Development Stage',
             risk_status = 'Risk status',
             ses = SES,
             ms_measure = 'MS Measure',
             ms_measure_description = 'MS Measure Description',
             mindset_type = 'Mindset Type',
             achievement_measure_description = 'Achievement Measure Description',
             academic_achievement_measure_type = 'Academic Achievement Measure Type',
             lab_based = 'Lab-based',
             published = 'Published',
             es_type = 'ES type',
             calculation = 'Calculation',
             variance = 'Variance',
             adjusted_variance = 'Adjusted Variance',
             is_significant = 'Significant?',
             growth_m = 'Growth M',
             growth_sd = 'Growth SD',
             other_m = 'Other M',
             other_sd = 'Other SD',
             cohen_d = "Cohen's d",
             calculated_r = 'Calculated r',
             notes = Notes)


# Check that variable types is correct
glimpse(df2)
```

```
## Rows: 273
## Columns: 35
## $ document_id                       <dbl> 1, 2, 2, 2, 3, 3, 3, 3, 4, 5, 6, ...
## $ study_id                          <dbl> 1, 2, 2, 2, 3, 3, 3, 3, 4, 5, 6, ...
## $ sample_id                         <dbl> 1, 2, 3, 4, 5, 163, 164, 165, 6, ...
## $ sample_country                    <chr> "Indonesia", "USA", "USA", "USA",...
## $ es_id                             <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11...
## $ reference                         <chr> "Adatitomo (2015)", "Bagley (2016...
## $ n                                 <dbl> 123, 400, 1019, 710, 250, 272, 27...
## $ adjusted_n                        <dbl> 123.000000, 400.000000, 1019.0000...
## $ student_description               <chr> "second semester university stude...
## $ school_level                      <chr> "post-secondary", "post-secondary...
## $ development_stage                 <chr> "Adults", "Adults", "Adults", "Ad...
## $ risk_status                       <chr> "low", "moderate", "moderate", "m...
## $ ses                               <chr> "not reported", "not reported", "...
## $ ms_measure                        <chr> "Mindset about intelligence", "Dw...
## $ ms_measure_description            <chr> "6 items, 3 growth and 3 fixed fr...
## $ mindset_type                      <chr> "Intelligence", "Personal attribu...
## $ achievement_measure_description   <chr> "Statistics final exam grade", "D...
## $ academic_achievement_measure_type <chr> "Course exam", "Course grade", "C...
## $ lab_based                         <chr> "no", "no", "no", "no", "no", "no...
## $ published                         <chr> "yes", "no", "no", "no", "no", "n...
## $ es_type                           <chr> "continuous", "continuous", "cont...
## $ calculation                       <chr> "Pearson's r", "sqrt of bivariate...
## $ variance                          <dbl> 0.0079425749, 0.0024188215, 0.000...
## $ adjusted_variance                 <dbl> 0.0079425749, 0.0024188215, 0.000...
```

```
## $ is_significant                   <chr> "N", "Y", "Y", "Y", "Y", "Y", "Y"...
## $ r                                <dbl> -0.12500000, 0.13266499, 0.197230...
## $ growth_m                         <dbl> NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ growth_sd                        <dbl> NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ other_m                          <dbl> NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ other_sd                         <dbl> NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ cohen_d                          <dbl> NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ rpb                              <dbl> NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ rb                               <dbl> NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ calculated_r                     <dbl> NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ notes                            <chr> "the authors of the study also me...
```

```r
# Change school_level from character to factor
df2$school_level <- as.factor(df2$school_level)
levels(df2$school_level)
```

```
##  [1] "elementary, middle and high"
##  [2] "graduate"
##  [3] "middle"
##  [4] "middle and secondary"
##  [5] "middle and secondary (mostly secondary)"
##  [6] "post-secondary"
##  [7] "primary"
##  [8] "primary and middle"
##  [9] "secondary"
## [10] "vocational courses"
```

```r
# Change development_stage from character to factor
df2$development_stage <- as.factor(df2$development_stage)
levels(df2$development_stage)
```

```
## [1] "Adolescents" "Adults"      "Children"    "Wide range"  "Wide Range"
```

```r
# Convert all "Wide range" level to "Wide Range"
df2$development_stage <- recode_factor(df2$development_stage,
                                       'Wide range' = 'Wide Range')
levels(df2$development_stage)
```

```
## [1] "Wide Range"  "Adolescents" "Adults"      "Children"
```

```r
# Change risk_status from character to factor
df2$risk_status <- as.factor(df2$risk_status)
levels(df2$risk_status)
```

```
## [1] "."        "high"     "low"      "moderate"
```

```r
## Note: The category '.' applies to 4 rows
## These are studies from which it was not possible to determine the risk status
df2 %>%
  filter(risk_status == '.')
```

```
## # A tibble: 4 x 35
##   document_id study_id sample_id sample_country es_id reference     n adjusted_n
##         <dbl>    <dbl>     <dbl> <chr>          <dbl> <chr>     <dbl>      <dbl>
## 1          31       32        45 USA               82 Ehrlinge~    95         95
## 2          32       33        46 USA               83 Ehriling~   100        100
## 3          33       34        47 USA               84 Ehrlinge~   122        122
## 4          34       35        48 USA               85 Ehriling~   122        122
```

```
## # ... with 27 more variables: student_description <chr>, school_level <fct>,
## #   development_stage <fct>, risk_status <fct>, ses <chr>, ms_measure <chr>,
## #   ms_measure_description <chr>, mindset_type <chr>,
## #   achievement_measure_description <chr>,
## #   academic_achievement_measure_type <chr>, lab_based <chr>, published <chr>,
## #   es_type <chr>, calculation <chr>, variance <dbl>, adjusted_variance <dbl>,
## #   is_significant <chr>, r <dbl>, growth_m <dbl>, growth_sd <dbl>,
## #   other_m <dbl>, other_sd <dbl>, cohen_d <dbl>, rpb <dbl>, rb <dbl>,
## #   calculated_r <dbl>, notes <chr>
```

```r
# Change ses from character to factor
df2$ses <- as.factor(df2$ses)
levels(df2$ses)
```

```
## [1] "low SES"      "not low"      "not reported"
```

```r
# Change mindset_type from character to factor
df2$mindset_type <- as.factor(df2$mindset_type)
levels(df2$mindset_type)
```

```
##  [1] "Ability"
##  [2] "Ability and Intelligence"
##  [3] "Ability and Performance"
##  [4] "Ability to learn"
##  [5] "Art Ability"
##  [6] "Biology Ability"
##  [7] "English Ability"
##  [8] "Intelligence"
##  [9] "Intelligence and Reading Ability"
## [10] "Intelligence and Talent"
## [11] "Intelligence, Math Ability, and Effort"
## [12] "Math ability"
## [13] "Math Ability"
## [14] "Math intelligence"
## [15] "Math Intelligence"
## [16] "Performance and Intelligence"
## [17] "Personal attributes"
## [18] "Personality"
## [19] "Physics Intelligence"
## [20] "Reading Ability"
## [21] "School Ability"
## [22] "Science ability"
## [23] "Science Ability"
## [24] "Talent for School"
## [25] "Verbal Intelligence"
```

```r
# Change academic_achievement_measure_type from character to factor
df2$academic_achievement_measure_type <- as.factor(df2$academic_achievement_measure_type)
levels(df2$academic_achievement_measure_type)
```

```
## [1] "Course exam"       "Course grade"      "GPA"
## [4] "Standardized test"
```

```r
# Change lab_based from character to factor
df2$lab_based <- as.factor(df2$lab_based)
levels(df2$lab_based)
```

```
## [1] "no"  "yes"
```

```r
# Change published from character to factor
df2$published <- as.factor(df2$published)
levels(df2$published)
```

```
## [1] "no"  "yes"
```

```r
# Change es_type from character to factor
df2$es_type <- as.factor(df2$es_type)
levels(df2$es_type)
```

```
## [1] "categorical" "continuous"
```

```r
# Change is_significant from character to factor
df2$is_significant <- as.factor(df2$is_significant)
levels(df2$is_significant)
```

```
## [1] "N" "Y"
```

Meta-analysis 1

- REM
- REML to estimate tau
- No hakn correction

```
meta1 <- metacor(cor = r,
                 n = n,
                 studlab = reference,
                 data= df2,
                 fixed = FALSE,
                 random = TRUE,
                 method.tau = 'REML',
                 hakn = FALSE,
                 title = "Mindset and Academic Achievement",
                 prediction = TRUE)
meta1
```

```
## Review:     Mindset and Academic Achievement
##
## Number of studies combined: k = 273
## Number of observations: o = 419854
##
##                           COR            95%-CI     z  p-value
## Random effects model 0.1067 [ 0.0867; 0.1265] 10.41 < 0.0001
## Prediction interval         [-0.1618; 0.3604]
##
## Quantifying heterogeneity:
##  tau^2 = 0.0187 [0.0164; 0.0284]; tau = 0.1369 [0.1279; 0.1685]
##  I^2 = 97.0% [96.8%; 97.1%]; H = 5.74 [5.56; 5.92]
##
## Test of heterogeneity:
##        Q d.f. p-value
##  8958.24  272       0
##
## Details on meta-analytical method:
## - Inverse variance method
## - Restricted maximum-likelihood estimator for tau^2
## - Q-profile method for confidence interval of tau^2 and tau
## - Fisher's z transformation of correlations
```

- 273 studies were combined in the meta-analysis
- Estimated correlation is 0.1070679 with standard error 0.0102841 with 95% confidence interval [0.0869115, 0.1272244], which excludes zero.

Analysis of between-studies heterogeneity

- Cochrane's Q: If there was no heterogeneity this statistics should be distributed as a χ^2 distribution with 272 degrees of freedom. In our meta-analysis $Q = $8958.2395056. We reject the null hypothesis of homogeneity. There is evidence for heterogeneity.
- I^2 = 0.9696369(95%CI:0.9676962-0.9676962%), meanining that about 97% of the variability in effect sizes is due between-study heterogeneity. This can be considered substantial heterogeneity (according to Thmppson's rule of thumb).
- H, the square root of H^2, is 5.7388766. Values greater than 1 indicate heterogeneity.
- τ^2, the between-study variance or 'variance of the true effect size underlying our data', is 0.0187471 with 95% confidence interval [0.0163586, 0.0283962], which does not include zero. Indicates heterogeneity. The confidence interval for τ^2 was calculated based on the Q-profile method.
- τ, is the 'standard deviation of the true effect size and [...] it tells us something about the range of the true effect sizes. The true effect sizes have an estimated standard deviation of $SD = $0.13692 expressed on the scale of correlation coefficient.
- The prediction interval ranges from r = -0.1632535 to 0.3773894. This means that it is possible that some future studies will find a negative correlation between mindset and academic achievement based on the present evidence. But the interval spans also over to a substantial positive effect.

Outliers and influential cases

Basic outlier removal

Function:

```
#' Find Statistical Outliers in a Meta-Analysis
#'
#' Searches for statistical outliers in meta-analysis results generated by \code{\link[meta]{meta}} fun
#' \code{\link[metafor]{rma.uni}} in the \code{metafor} package.
#'
#' @usage find.outliers(x, ...)
#'
#' @param x Either (1) an object of class \code{meta}, generated by the \code{metabin}, \code{metagen},
#' \code{metacont}, \code{metacor}, \code{metainc}, \code{metarate} or \code{metaprop} function; or (2)
#' and object of class \code{rma.uni} created with the \code{\link[metafor]{rma.uni}} function in \code
#' @param ... Additional parameters for the \code{\link[metafor]{rma.uni}} or \code{\link[meta]{update.
#'
#' @details
#' This function searches for outlying studies in a meta-analysis results object. Studies are defined a
#' their 95\% confidence interval lies ouside the 95\% confidence interval of the pooled effect.
#'
#' When outliers are found, the function automatically recalculates the meta-analysis results, using th
#' in the object provided in \code{x}, but excluding the detected outliers.
#'
#' A forest plot of the meta-analysis with outliers removed can be generated directly by plugging the o
#' the \code{forest} function.
#'
#' @references Harrer, M., Cuijpers, P., Furukawa, T.A, & Ebert, D. D. (2019).
#' \emph{Doing Meta-Analysis in R: A Hands-on Guide}. DOI: 10.5281/zenodo.2551803. \href{https://bookdo
#'
#' @author Mathias Harrer & David Daniel Ebert
#'
#' @return
#' Returns the identified outliers and the meta-analysis results when the outliers are removed.
#'
#' If the provided meta-analysis object is of class \code{meta}, the following objects are returned if
#' results of the function are saved to another object:
#' \itemize{
#' \item \code{out.study.fixed}: A numeric vector containing the names of the outlying studies when
#' assuming a fixed-effect model.
#' \item \code{out.study.random}: A numeric vector containing the names of the outlying studies when
#' assuming a random-effects model. The \eqn{\tau^{2}} estimator \code{method.tau} is inherited from \c
#' \item \code{m.fixed}: An object of class \code{meta} containing the results of the meta-analysis wit
#' removed (assuming a fixed-effect model).
#' \item \code{m.random}: An object of class \code{meta} containing the results of the meta-analysis wi
#' removed (assuming a random-effects model, and using the same \code{method.tau} as in the original an
#'}
#'
#' If the provided meta-analysis object is of class \code{rma.uni}, the following objects are returned
#' results of the function are saved to another object:
#' \itemize{
#' \item \code{out.study}: A numeric vector containing the names of the outlying studies.
#' \item \code{m}: An object of class \code{rma.uni} containing the results of the meta-analysis with o
#' removed (using the same settings as in the meta-analysis object provided).
```

```
#'}
#' @importFrom metafor rma.uni
#' @importFrom meta update.meta
#'
#' @export find.outliers
#'
#' @aliases spot.outliers.random spot.outliers.fixed spot.outliers
#'
#' @seealso \code{\link[metafor]{influence.rma.uni}}, \code{\link[meta]{metainf}}, \code{\link[meta]{ba
#'
#' @examples
#' suppressPackageStartupMessages(library(meta))
#' suppressPackageStartupMessages(library(metafor))
#' suppressPackageStartupMessages(library(dmetar))
#'
#' # Pool with meta
#' m1 <- metagen(TE, seTE, data = ThirdWave,
#'               studlab = ThirdWave$Author, comb.fixed = FALSE)
#'
#' # Pool with metafor
#' m2 <- rma(yi = TE, sei = seTE, data = ThirdWave,
#'           slab = ThirdWave$Author, method = "PM")
#'
#' # Find outliers
#' fo1 <- find.outliers(m1)
#' fo2 <- find.outliers(m2)
#'
#' # Show summary
#' summary(fo1)
#' summary(fo2)
#'
#' \dontrun{
#' # Make forest plot
#' # Pass additional arguments from meta & metafor's forest function
#' forest(fo1, prediction = TRUE)
#' forest(fo2, cex = .8, col = "lightblue")
#' }


find.outliers = spot.outliers.random = spot.outliers.fixed = function(x, ...){


  if (class(x)[1] %in% c("rma.uni", "rma")){

    token = "metafor"

    # Generate lower/upper for all effects
    lower = as.numeric(x$yi - 1.96*sqrt(x$vi))
    upper = as.numeric(x$yi + 1.96*sqrt(x$vi))

    # Select outliers
    mask = upper < x$ci.lb | lower > x$ci.ub
    dat = data.frame("yi" = x$yi[!mask],
```

```r
                     "vi" = x$vi[!mask],
                     "studlab" = as.character(x$slab[!mask]))
  out.study = x$slab[mask]

  # Update metafor model
  method.tau = x$method
  m = metafor::rma.uni(dat$yi, vi = dat$vi, method = method.tau, slab = dat$studlab, ...)

  if (length(out.study) < 1){

    tau.token = "metafor.null"
    cat(paste0("No outliers detected (", method.tau,")."))
    out.study = NULL

  } else {

    tau.token = "metafor"

  }

}

if (class(x)[1] %in% c("metagen", "metapropr",
                         "metacor", "metainc", "metacont",
                         "metaprop", "metabin", "metabin")){

  token = "meta"

  # Control for objects with NAs in study data
  if (anyNA(x$TE) | anyNA(x$seTE)){
    warning("Studies with NAs not considered in outlier analysis.")

  }

  if (class(x)[1] == "metaprop"){

    lower = x$TE - 1.96*x$seTE
    upper = x$TE + 1.96*x$seTE

    # Generate mask with outliers (fixed/random)
    mask.fixed =
      (!is.na(upper) & upper < x$lower.fixed) |
      (!is.na(lower) & lower > x$upper.fixed)
    mask.random =
      (!is.na(upper) & upper < x$lower.random) |
      (!is.na(lower) & lower > x$upper.random)

  } else {

    # Generate mask with outliers (fixed/random)
    mask.fixed =
      (!is.na(x$upper) & x$upper < x$lower.fixed) |
      (!is.na(x$lower) & x$lower > x$upper.fixed)
```

```
    mask.random =
      (!is.na(x$upper) & x$upper < x$lower.random) |
      (!is.na(x$lower) & x$lower > x$upper.random)

  }

  # Update meta-analysis with outliers removed
  m.fixed = update.meta(x, exclude = mask.fixed, ...)
  m.random = update.meta(x, exclude = mask.random, ...)

  # Select names of outlying studies
  out.study.fixed = x$studlab[mask.fixed]
  out.study.random = x$studlab[mask.random]

  if (x$comb.fixed == TRUE & x$comb.random == FALSE){

    if (length(out.study.fixed) < 1){

      tau.token = "null.ftrf"
      out.study.fixed = NULL

    } else {

      tau.token = "ftrf"

    }

  }

  if (x$comb.fixed == FALSE & x$comb.random == TRUE){

    if (length(out.study.random) < 1){

      tau.token = "null.ffrt"
      out.study.random = NULL

    } else {

      tau.token = "ffrt"

    }

  }

  if (x$comb.fixed == TRUE & x$comb.random == TRUE){

    if (length(out.study.fixed) < 1 & length(out.study.random) < 1){

      out.study.fixed = NULL
      out.study.random = NULL
      tau.token = "null.ftrt"

    } else {
```

```
        if (length(out.study.fixed) < 1){out.study.fixed = NULL}
        if (length(out.study.random) < 1){out.study.random = NULL}

        tau.token = "ftrt"

      }

    }

  }

  if (!class(x)[1] %in% c("rma.uni", "rma", "metacont",
                          "metagen", "metapropr",
                          "metacor", "metainc",
                          "metaprop", "metabin", "metabin")){

    message("Input must be of class 'meta' or 'rma.uni'")

  }

  if (token == "metafor"){

    returnlist = list("out.study" = out.study,
                      "m" = m)

    if (tau.token == "metafor"){class(returnlist) = c("find.outliers", "mf", method.tau)}
    if (tau.token == "metafor.null"){class(returnlist) = c("find.outliers", "mf.null", method.tau)}

    # Return
    invisible(returnlist)

    returnlist

  } else {

    returnlist = list("out.study.fixed" = out.study.fixed,
                      "out.study.random" = out.study.random,
                      "m.fixed" = m.fixed,
                      "m.random" = m.random)

    # Set classes
    if (tau.token == "ftrf"){class(returnlist) = c("find.outliers", "ftrf")}
    if (tau.token == "ffrt"){class(returnlist) = c("find.outliers", "ffrt")}
    if (tau.token == "ftrt"){class(returnlist) = c("find.outliers", "ftrt")}
    if (tau.token == "null.ftrf"){class(returnlist) = c("find.outliers", "null.ftrf")}
    if (tau.token == "null.ffrt"){class(returnlist) = c("find.outliers", "null.ffrt")}
    if (tau.token == "null.ftrt"){class(returnlist) = c("find.outliers", "null.ftrt")}

    # Return
    invisible(returnlist)

    returnlist
```

```r
  }

}
```

```r
find.outliers(meta1)
```

```
## $out.study.fixed
##   [1] "Adatitomo (2015)"
##   [2] "Bagley (2016) - S1"
##   [3] "Bagley (2016) - S2"
##   [4] "Bagley (2016) - S3"
##   [5] "Benningfield (2013) - S1"
##   [6] "Benningfield (2013) - S2"
##   [7] "Bergen (1991)"
##   [8] "Black (2008) - S1 M1"
##   [9] "Black (2008) - S1 M2"
##  [10] "Black (2008) - S3 M1"
##  [11] "Black (2008) - S4 M1"
##  [12] "Black (2008) - S5 M1"
##  [13] "Blackwell et al. (2007) - S1 M1"
##  [14] "Blackwell et al. (2007) - S1 M2"
##  [15] "Blackwell et al. (2007) - S2 M1"
##  [16] "Boazman (2010) - S1 M1"
##  [17] "Boazman (2010) - S1 M2"
##  [18] "Boazman (2010) - S2 M1"
##  [19] "Broome (2001) - S2"
##  [20] "Cadwallader (2009)"
##  [21] "Callahan et al. (2015)"
##  [22] "Chen (2012) - M2"
##  [23] "Chen & Wong (2015)"
##  [24] "Claro et al. (2016)"
##  [25] "Clevenger (2013)"
##  [26] "Cordell-McNulty (2009)"
##  [27] "Da Fonseca (2009) - M1"
##  [28] "Dai & Cromley (2014) - M2"
##  [29] "De Castella & Byrne (2015)"
##  [30] "Dickhäuser et al. (2016)"
##  [31] "Dinger et al. (2013)"
##  [32] "Diseth et al. (2014) - M1"
##  [33] "Edwards (2014)"
##  [34] "Ehrlinger (2016)"
##  [35] "Ehrilinger & Brewer (2016)"
##  [36] "Ehrlinger & Conlon (2016)"
##  [37] "Ehrilinger & Dweck (2016)"
##  [38] "Ehrlinger, Mitchum et al. (2016)"
##  [39] "Eskreis-Winkler et al. (2016) - S1 M1"
##  [40] "Eskreis-Winkler et al. (2016) - S1 M2"
##  [41] "Faria & Fontaine (1997) - M1"
##  [42] "Feldman et al. (2016)"
##  [43] "Fillmore (2015)"
##  [44] "Flanigan et al. (2015) - M1"
##  [45] "Flanigan et al. (2015) - M2"
##  [46] "Fleming (2007)"
##  [47] "Froehlich et al. (2016) - S5"
```

```
##  [48] "Froehlich et al. (2016) - S6"
##  [49] "Froehlich et al. (2016) - S7"
##  [50] "Furnham et al. (2002)"
##  [51] "Gaultney (1989) - S3 M3"
##  [52] "Greene et al. (2015)"
##  [53] "Gubi (2012) - M1"
##  [54] "Guich (2007) - M2"
##  [55] "Harpalani (2005)"
##  [56] "Hazard (1997) - M1"
##  [57] "Hazard (1997) - M2"
##  [58] "Hendricks (2012)"
##  [59] "Holden et al. (2016)"
##  [60] "Hotulainen & Telivuo (2014)"
##  [61] "Howell (2009) - M1"
##  [62] "Howell (2009) - M2"
##  [63] "Hwang et al. (2016) - M1"
##  [64] "Hwang et al. (2016) - M2"
##  [65] "Kennett & Keefer (2006)"
##  [66] "King (2012) - M2"
##  [67] "Kornilova et al. (2009)"
##  [68] "Kornilova et al. (2015) - S1"
##  [69] "Kornilova et al. (2015) - S2"
##  [70] "Leondari & Gialamas (2002)"
##  [71] "Lewis (1998)"
##  [72] "Lindsay (2006)"
##  [73] "Linehan (1998) - M2"
##  [74] "Luo et al. (2014) - M1"
##  [75] "Luo et al. (2014) - M2"
##  [76] "Luo et al. (2014) - M4"
##  [77] "Macdonald (2016)"
##  [78] "Magno (2012) - M1"
##  [79] "Magno (2012) - M2"
##  [80] "Martin et al. (2013) - M1"
##  [81] "Martin et al. (2013) - M2"
##  [82] "Matheson (2015)"
##  [83] "Miller (2010) - M1"
##  [84] "Miller (2010) - M2"
##  [85] "Moser et al. (2013)"
##  [86] "Northrop (2014) - M3"
##  [87] "P'Pool (2012) - M1"
##  [88] "P'Pool (2012) - M2"
##  [89] "Ravenscroft et al. (2012)"
##  [90] "Renaud-Dube (2015)"
##  [91] "Rheinschmidt & Mendoza-Denton (2014) - S1"
##  [92] "Rheinschmidt & Mendoza-Denton (2014) - S2 M2"
##  [93] "Rheinschmidt & Mendoza-Denton (2014) - S3 M2"
##  [94] "Riyaz (2013) - M1"
##  [95] "Riyaz (2013) - M2"
##  [96] "Riyaz (2013) - M3"
##  [97] "Riyaz (2013) - M4"
##  [98] "Riyaz (2013) - M5"
##  [99] "Riyaz (2013) - M6"
## [100] "Riyaz (2013) - M7"
## [101] "Robins & Pals (2002) - M1"
```

```
## [102] "Robins & Pals (2002) - M2"
## [103] "Rudig (2014)"
## [104] "Ryan et al. (2007) - M1"
## [105] "Ryan et al. (2007) - M2"
## [106] "Schnedecker (1997) - M1"
## [107] "Schnedecker (1997) - M2"
## [108] "Schroder et al. (2016) - M1"
## [109] "Schroder et al. (2016) - M2"
## [110] "Schullo (1996) - S4 M1"
## [111] "Schwinger et al. (2016)"
## [112] "Shedlosky-Shoemaker & Fautch (2015)"
## [113] "Shell et al. (2016) - M1"
## [114] "Shell et al. (2016) - M2"
## [115] "Shell et al. (2016) - M3"
## [116] "Shell et al. (2016) - M4"
## [117] "Shively & Ryan (2013) - S1"
## [118] "Stipek & Gralinski (1996) - M5"
## [119] "Stipek & Gralinski (1996) - M6"
## [120] "Stipek & Gralinski (1996) - M7"
## [121] "Stipek & Gralinski (1996) - M8"
## [122] "Stump et al. (2014) - M1"
## [123] "Stump et al. (2014) - M2"
## [124] "Tallman (2000)"
## [125] "Tarbetsky et al. (2016) - S1"
## [126] "Tempelaar et al. (2015) - M1"
## [127] "Tempelaar et al. (2015) - M2"
## [128] "Tempelaar et al. (2015) - M3"
## [129] "Tucker-Drob et al. (2016) - M1"
## [130] "Tucker-Drob et al. (2016) - M2"
## [131] "Uchida (2004)"
## [132] "West (2016)"
## [133] "West et al. (2016) - M2"
## [134] "Yeager et al. (2016)"
## [135] "Ziegler et al. (2010)"
## [136] "Ziegler & Stoeger (2010)"
## [137] "Zientek et al. (2013)"
##
## $out.study.random
##  [1] "Adatitomo (2015)"
##  [2] "Bagley (2016) - S2"
##  [3] "Benningfield (2013) - S3"
##  [4] "Bergen (1991)"
##  [5] "Bettinger et al. (submitted)"
##  [6] "Black (2008) - S3 M1"
##  [7] "Black (2008) - S3 M3"
##  [8] "Black (2008) - S4 M1"
##  [9] "Black (2008) - S7 M1"
## [10] "Broome (2001) - S2"
## [11] "Cadwallader (2009)"
## [12] "Cain et al. (2016) - M1"
## [13] "Cain et al. (2016) - M2"
## [14] "Callahan et al. (2015)"
## [15] "Chen (2012) - M1"
## [16] "Chen & Pajaras (2010) - M1"
```

```
## [17] "Chen & Pajaras (2010) - M2"
## [18] "Claro et al. (2016)"
## [19] "Clevenger (2013)"
## [20] "Cordell-McNulty (2009)"
## [21] "Cury et al. (2006) - M1"
## [22] "Cury et al. (2006) - M2"
## [23] "Da Fonseca (2009) - M2"
## [24] "Delavar et al. (2011) - M1"
## [25] "Dinger et al. (2013)"
## [26] "Diseth et al. (2014) - M2"
## [27] "Ehrlinger (2016)"
## [28] "Ehrilinger & Brewer (2016)"
## [29] "Ehrilinger & Dweck (2016)"
## [30] "Ehrlinger, Hartwig et al. (2016a)"
## [31] "Ehrlinger, Hartwig et al. (2016b)"
## [32] "Eskreis-Winkler et al. (2016) - S2 M2"
## [33] "Faria & Fontaine (1997) - M2"
## [34] "Feldman et al. (2016)"
## [35] "Flanigan et al. (2015) - M1"
## [36] "Flanigan et al. (2015) - M2"
## [37] "Froehlich et al. (2016) - S1"
## [38] "Froehlich et al. (2016) - S5"
## [39] "Froehlich et al. (2016) - S7"
## [40] "Gaultney (1989) - S1 M2"
## [41] "Gaultney (1989) - S1 M3"
## [42] "Greene et al. (2015)"
## [43] "Gubi (2012) - M1"
## [44] "Haimovitz et al. (2011)"
## [45] "Hazard (1997) - M3"
## [46] "Hendricks (2012)"
## [47] "Holden et al. (2016)"
## [48] "Hwang et al. (2016) - M1"
## [49] "Hwang et al. (2016) - M2"
## [50] "King (2012) - M1"
## [51] "Kornilova et al. (2009)"
## [52] "Kornilova et al. (2015) - S2"
## [53] "Kraft & Grace (2016) - M1"
## [54] "Kraft & Grace (2016) - M2"
## [55] "Law (2009)"
## [56] "Lewis (1998)"
## [57] "Linehan (1998) - M1"
## [58] "Linehan (1998) - M3"
## [59] "Luo et al. (2014) - M1"
## [60] "Luo et al. (2014) - M2"
## [61] "Luo et al. (2014) - M3"
## [62] "Macdonald (2016)"
## [63] "Magno (2012) - M1"
## [64] "Martin et al. (2013) - M2"
## [65] "Matheson (2015)"
## [66] "Moser et al. (2013)"
## [67] "P'Pool (2012) - M1"
## [68] "P'Pool (2012) - M2"
## [69] "Rheinschmidt & Mendoza-Denton (2014) - S1"
## [70] "Riley (2003)"
```

```
## [71] "Robins & Pals (2002) - M1"
## [72] "Robins & Pals (2002) - M2"
## [73] "Romero et al. (2014)"
## [74] "Schroder et al. (2016) - M2"
## [75] "Schullo (1996) - S4 M1"
## [76] "Shedlosky-Shoemaker & Fautch (2015)"
## [77] "Shih (2007) - M1"
## [78] "Stump et al. (2014) - M2"
## [79] "Tarbetsky et al. (2016) - S1"
## [80] "Tempelaar et al. (2015) - M1"
## [81] "Tempelaar et al. (2015) - M2"
## [82] "Tempelaar et al. (2015) - M3"
## [83] "Volpe (2016) - M2"
## [84] "Volpe (2016) - M3"
## [85] "West (2016)"
## [86] "West et al. (2016) - M1"
## [87] "West et al. (2016) - M2"
## [88] "Yeager et al. (2016)"
## [89] "Zhao & Wang (2014)"
##
## $m.fixed
## Review:     Mindset and Academic Achievement
##
## Number of studies combined: k = 136
## Number of observations: o = 419854
##
##                         COR            95%-CI     z  p-value
## Random effects model 0.2424 [0.2298; 0.2550] 36.16 < 0.0001
## Prediction interval         [0.1998; 0.2841]
##
## Quantifying heterogeneity:
##  tau^2 = 0.0005 [0.0000; 0.0031]; tau = 0.0216 [0.0000; 0.0558]
##  I^2 = 8.0% [0.0%; 26.5%]; H = 1.04 [1.00; 1.17]
##
## Test of heterogeneity:
##       Q d.f. p-value
##  146.74  135  0.2312
##
## Details on meta-analytical method:
## - Inverse variance method
## - Restricted maximum-likelihood estimator for tau^2
## - Q-profile method for confidence interval of tau^2 and tau
## - Fisher's z transformation of correlations
##
## $m.random
## Review:     Mindset and Academic Achievement
##
## Number of studies combined: k = 184
## Number of observations: o = 419854
##
##                         COR            95%-CI     z  p-value
## Random effects model 0.1046 [0.0916; 0.1175] 15.66 < 0.0001
## Prediction interval         [0.0252; 0.1826]
##
```

```
## Quantifying heterogeneity:
##  tau^2 = 0.0016 [0.0006; 0.0049]; tau = 0.0399 [0.0238; 0.0703]
##  I^2 = 26.8% [11.7%; 39.4%]; H = 1.17 [1.06; 1.28]
##
## Test of heterogeneity:
##       Q d.f. p-value
##  250.11  183  0.0007
##
## Details on meta-analytical method:
## - Inverse variance method
## - Restricted maximum-likelihood estimator for tau^2
## - Q-profile method for confidence interval of tau^2 and tau
## - Fisher's z transformation of correlations
##
## attr(,"class")
## [1] "find.outliers" "ffrt"
```

- 137 were identified: from 273 included studies, the anlaysis includes only 136
- I^2 moves from about 97% to 26.8%
- tau^2 is now closer to zero tau^2 = 0.0016 [0.0006; 0.0049]
- The Q statistics is still significant: Q = 250.11, df = 183, p-value = 0.0007
- The estimated effect does not change substantialli: r = 0.1046, 95% CI [0.0916; 0.1175]

Influence analysis

Function:

```
#' Influence Diagnostics
#'
#' Conducts an influence analysis of a meta-analysis generated by \code{\link[meta]{meta}} functions,
#' and allows to produce influence diagnostic plots.
#'
#' @usage InfluenceAnalysis(x, random = FALSE, subplot.heights = c(30,18),
#'     subplot.widths = c(30,30), forest.lims = 'default',
#'     return.separate.plots = FALSE, text.scale = 1)
#'
#' @param x An object of class \code{meta}, generated by the \code{metabin}, \code{metagen},
#' \code{metacont}, \code{metacor}, \code{metainc}, \code{metarate} or \code{metaprop} function.
#' @param random Logical. Should the random-effects model be used to generate the influence diagnostics
#' Uses the \code{method.tau} specified in the \code{meta} object if one
#' of "\code{DL}", "\code{HE}", "\code{SJ}", "\code{ML}", "\code{REML}", "\code{EB}", "\code{PM}", "\co
#' the \code{\link[metafor]{metafor}} package). Otherwise, the DerSimonian-Laird
#' (\code{"DL"}; DerSimonian & Laird, 1986) estimator is used. \code{FALSE} by default.
#' @param subplot.heights Concatenated array of two numerics. Specifies the heights of the
#' first (first number) and second (second number) row of the overall plot generated when plotting the
#' Default is \code{c(30,18)}.
#' @param subplot.widths Concatenated array of two numerics. Specifies the widths of the
#' first (first number) and second (second number) column of the overall results plot generated when pl
#' Default is \code{c(30,30)}.
#' @param forest.lims Concatenated array of two numerics. Specifies the x-axis limits of the forest plo
#' generated when plotting the results. Use \code{"default"} if standard settings should be used (this
#' @param return.separate.plots Logical. When plotted, should the influence plots be shown as separate
#' of returning them in one overall plot?
#' @param text.scale Positive numeric. Scaling factor for the text geoms used when plotting the results
#' text, while values >1 increase the text size. Default is \code{1}.
#'
#' @details
#' The function conducts an influence analysis using the "Leave-One-Out" paradigm internally
#' and produces data for four influence diagnostics. Diagnostic plots can be produced by saving the out
#' function to an object and plugging it into the \code{plot} function.
#' These diagnostics may be used to determine which study or effect size
#' may have an excessive influence on the overall results of a meta-analysis and/or contribute substant
#' the between-study heterogeneity in an analysis. This may be used for outlier detection and to test
#' the robustness of the overall results found in an analysis. Results for four diagnostics are calcula
#' \itemize{
#' \item \strong{Baujat Plot}: Baujat et al. (2002) proposed a plot to evaluate heterogeneity patterns
#' a meta-analysis. The x-axis of the Baujat plot shows the overall heterogeneity contribution of each
#' while the y-axis shows the influence of each effect size on the pooled result. The \code{\link[meta]
#' function is called internally to produce the results. Effect sizes or studies with high values
#' on both the x and y-axis may be considered to be influential cases; effect sizes or studies
#' with high heterogeneity contribution (x-axis) and low influence on the overall results can be outlie
#' which might be deleted to reduce the amount of between-study heterogeneity.
#' \item \strong{Influence Characteristics}: Several influence analysis diagnostics
#' proposed by Viechtbauer & Cheung (2010). Results are calculated by an internal call
#' to \code{\link[metafor]{influence.rma.uni}}. In the console output, potentially influential studies
#' with an asterisk (\code{*}). When plotted, effect sizes/studies determined to be influential cases
#' using the "rules of thumb" described in Viechtbauer & Cheung (2010) are shown in red. For further
#' details, see the documentation of the \code{\link[metafor]{influence.rma.uni}} function.
```

```
#' \item \strong{Forest Plot for the Leave-One-Out Analysis, sorted by Effect Size}: This
#' displays the effect size and \eqn{I^2}-heterogeneity when omitting one of the \eqn{k} studies each t
#' The plot is sorted by effect size to determine which studies or effect sizes particularly
#' affect the overall effect size. Results are generated by an internal call to \code{\link[meta]{metai
#' \item \strong{Forest Plot for the Leave-One-Out Analysis, sorted by \eqn{I^2}}: see above; results a
#' by \eqn{I^2} to determine the study for which exclusion results in the greatest reduction of heterog
#'}
#'
#' @references Harrer, M., Cuijpers, P., Furukawa, T.A, & Ebert, D. D. (2019).
#' \emph{Doing Meta-Analysis in R: A Hands-on Guide}. DOI: 10.5281/zenodo.2551803. \href{https://bookdo
#'
#' DerSimonian R. & Laird N. (1986), Meta-analysis in clinical trials. \emph{Controlled Clinical Trials
#'
#' Viechtbauer, W., & Cheung, M. W.-L. (2010). Outlier and influence diagnostics for meta-analysis. \em
#'
#' @author Mathias Harrer & David Daniel Ebert
#'
#' @return A \code{list} object of class \code{influence.analysis} containing the
#' following objects is returned (if results are saved to a variable):
#' \itemize{
#' \item \code{BaujatPlot}: The Baujat plot
#' \item \code{InfluenceCharacteristics}: The Viechtbauer-Cheung influence characteristics plot
#' \item \code{ForestEffectSize}: The forest plot sorted by effect size
#' \item \code{ForestI2}: The forest plot sorted by between-study heterogeneity
#' \item \code{Data}: A \code{data.frame} containing the data used for plotting.
#'}
#' Otherwise, the function prints out (1) the results of the Leave-One-Out Analysis (sorted by \eqn{I^2
#' (2) the Viechtbauer-Cheung Influence Diagnostics and  (3) Baujat Plot data (sorted by heterogeneity
#' in this order. Plots can be produced manually by plugging a saved object of class \code{InfluenceAna
#' the function into the \code{plot} function. It is also possible to only produce one specific plot by
#' specifying the name of the plot as a \code{character} in the second argument of the \code{plot} call
#'
#' @import ggplot2 ggrepel grid
#' @importFrom gridExtra grid.arrange arrangeGrob
#' @importFrom metafor rma.uni influence.rma.uni
#' @importFrom meta metainf
#' @importFrom graphics abline axis lines mtext par plot points rect segments text
#' @importFrom stats as.formula hat influence ks.test optimize pbinom pchisq pf pnorm pt punif qchisq q
#'
#' @export InfluenceAnalysis
#'
#' @seealso \code{\link[metafor]{influence.rma.uni}}, \code{\link[meta]{metainf}}, \code{\link[meta]{ba
#'
#' @examples
#'
#' \dontrun{
#' # Load 'ThirdWave' data
#' data(ThirdWave)
#'
#' # Create 'meta' meta-analysis object
#' suppressPackageStartupMessages(library(meta))
#' meta = metagen(TE, seTE, studlab = paste(ThirdWave$Author), data=ThirdWave)
#'
```

```r
#' # Run influence analysis; specify to return separate plots when plotted
#' inf.an = InfluenceAnalysis(meta, return.separate.plots = TRUE)
#'
#' # Show results in console
#' inf.an
#'
#' # Generate all plots
#' plot(inf.an)
#'
#' # For baujat plot
#' plot(inf.an, "baujat")
#'
#' # For influence diagnostics plot
#' plot(inf.an, "influence")
#'
#' # For forest plot sorted by effect size
#' plot(inf.an, "ES")
#'
#' # For forest plot sorted by I-squared
#' plot(inf.an, "I2")}


### Influence Analysis function for fixed-effect-model meta-analyses

InfluenceAnalysis = function(x, random = FALSE, subplot.heights = c(30, 18),
                             subplot.widths = c(30, 30),
                             forest.lims = "default", return.separate.plots = FALSE,
                             text.scale = 1) {

    # Validate
    x = x
    if (class(x)[1] %in% c("meta", "metabin", "metagen", "metacont", "metacor", "metainc", "metaprop",
      x <- update(x, subset = !(is.na(x$TE) | is.na(x$seTE)))
    } else {

        stop("Object 'x' must be of class 'meta', 'metabin', 'metagen', 'metacont', 'metacor', 'metainc
    }

    n.studies = x$k
    TE = x$TE
    seTE = x$seTE
    random = random

    # Make unique studlabs
    x$studlab = make.unique(x$studlab)

    if (random %in% c(TRUE, FALSE)) {

    } else {

        stop("'random' must be set to either TRUE or FALSE.")
    }
```

```
forest.lims = forest.lims

if (forest.lims[1] == "default" | (class(forest.lims[1]) == "numeric" & class(forest.lims[2]) == "nu

} else {
    stop("'forest.lims' must either be 'default' or two concatenated numerics for ymin and ymax.")
}

return.seperate.plots = return.separate.plots

if (return.seperate.plots %in% c(TRUE, FALSE)) {

} else {

    stop("'return.separate.plots' must be set to either TRUE or FALSE.")
}

heights = subplot.heights
if (class(heights[1]) == "numeric" & class(heights[2]) == "numeric") {

} else {
    stop("'subplot.heights' must be two concatenated numerics.")
}

widths = subplot.widths
if (class(widths[1]) == "numeric" & class(widths[2]) == "numeric") {

} else {
    stop("'widths' must be two concatenated numerics.")
}

text.scale = text.scale
if (text.scale > 0) {

} else {
    stop("'text.scale' must be a single number greater 0.")
}

if (length(unique(x$studlab)) != length(x$studlab)) {
    stop("'Study labels in the 'meta' object must be unique.")
}

########################################
cat("[===============")
########################################

if (random == FALSE) {
    method.rma = "FE"
    method.meta = "fixed"
} else {
    if (x$method.tau %in% c("DL", "HE", "SJ", "ML", "REML", "EB", "HS", "GENQ", "PM")) {
        method.rma = x$method.tau
    } else {
```

```
        method.rma = "DL"
        cat("Tau estimator is unkown to metafor::rma; DerSimonian-Laird ('DL') estimator used.")
    }
    method.meta = "random"
}

# Perform Meta-Analysis using metafor, get influence results
res = metafor::rma.uni(yi = TE, sei = seTE, measure = "GEN", data = x, method = method.rma, slab = s
metafor.inf = influence(res)
# Recode inf
metafor.inf$is.infl = ifelse(metafor.inf$is.infl == TRUE, "yes", "no")
cheungviechtdata = cbind(study = substr(rownames(as.data.frame(metafor.inf$inf)), 1, 3), as.data.fra
rownames(cheungviechtdata) = NULL

if (length(unique(cheungviechtdata$study)) < length(cheungviechtdata$study)) {

    i = 3

    while (length(unique(cheungviechtdata$study)) < length(cheungviechtdata$study)) {

        i = i + 1
        cheungviechtdata$study = substr(rownames(as.data.frame(metafor.inf$inf)), 1, i)

    }

}

# If study labels are only numeric: reset level indexing
if (sum(grepl("[A-Za-z]", levels(as.factor(cheungviechtdata$study)), perl = T)) == 0){

  cheungviechtdata$study = factor(cheungviechtdata$study, levels = sort(as.numeric(levels(cheungviec

}

########################################
cat("===============")
########################################

scalefun = function(x) sprintf("%.1f", x)

cheungviechtdata = as.data.frame(cheungviechtdata)

# Generate plots
rstudent.plot = ggplot2::ggplot(cheungviechtdata, aes(y = rstudent, x = study, color = is.infl, grou
    geom_line(color = "black") + geom_point(size = 2) + scale_color_manual(values = c("blue", "red")
    theme_minimal() + theme(axis.title.x = element_blank(), legend.position = "none", axis.text.x =
    size = 5), axis.title.y = element_text(size = 7), axis.text.y = element_text(size = 5)) + ylab(
    scale_y_continuous(labels = scalefun)

dffits.thresh = 3 * sqrt(metafor.inf$p/(metafor.inf$k - metafor.inf$p))
dffits.plot = ggplot2::ggplot(cheungviechtdata, aes(y = dffits, x = study, color = is.infl, group =
    geom_line(color = "black") + geom_point(size = 2) + scale_color_manual(values = c("blue", "red")
    theme_minimal() + theme(axis.title.x = element_blank(), legend.position = "none", axis.text.x =
```

```
        size = 5), axis.title.y = element_text(size = 7), axis.text.y = element_text(size = 5)) + ylab(
        scale_y_continuous(labels = scalefun)
    # geom_hline(yintercept = dffits.thresh, linetype='dashed', color='black')

    cook.d.plot = ggplot2::ggplot(cheungviechtdata, aes(y = cook.d, x = study, color = is.infl, group =
        geom_line(color = "black") + geom_point(size = 2) + scale_color_manual(values = c("blue", "red")
        theme_minimal() + theme(axis.title.x = element_blank(), legend.position = "none", axis.text.x =
        size = 5), axis.title.y = element_text(size = 7), axis.text.y = element_text(size = 5)) + ylab(
        scale_y_continuous(labels = scalefun)

    cov.r.plot = ggplot2::ggplot(cheungviechtdata, aes(y = cov.r, x = study, color = is.infl, group = 1)
        geom_line(color = "black") + geom_point(size = 2) + scale_color_manual(values = c("blue", "red")
        theme_minimal() + theme(axis.title.x = element_blank(), legend.position = "none", axis.text.x =
        size = 5), axis.title.y = element_text(size = 7), axis.text.y = element_text(size = 5)) + ylab(
        scale_y_continuous(labels = scalefun)

    tau2.del.plot = ggplot2::ggplot(cheungviechtdata, aes(y = tau2.del, x = study, color = is.infl, grou
        geom_line(color = "black") + geom_point(size = 2) + scale_color_manual(values = c("blue", "red")
        theme_minimal() + theme(axis.title.x = element_blank(), legend.position = "none", axis.text.x =
        size = 5), axis.title.y = element_text(size = 7), axis.text.y = element_text(size = 5)) + ylab(
        scale_y_continuous(labels = scalefun)

    QE.del.plot = ggplot2::ggplot(cheungviechtdata, aes(y = QE.del, x = study, color = is.infl, group =
        geom_line(color = "black") + geom_point(size = 2) + scale_color_manual(values = c("blue", "red")
        theme_minimal() + theme(axis.title.x = element_blank(), legend.position = "none", axis.text.x =
        size = 5), axis.title.y = element_text(size = 7), axis.text.y = element_text(size = 5)) + ylab(
        scale_y_continuous(labels = scalefun)

    hat.thresh = 3 * (metafor.inf$p/metafor.inf$k)
    hat.plot = ggplot2::ggplot(cheungviechtdata, aes(y = hat, x = study, color = is.infl, group = 1)) +
        geom_point(size = 2) + scale_color_manual(values = c("blue", "red")) + theme_minimal() + theme(a
        legend.position = "none", axis.text.x = element_text(angle = 45, size = 5), axis.title.y = eleme
        axis.text.y = element_text(size = 5)) + ylab("hat") + scale_y_continuous(labels = scalefun)
    # geom_hline(yintercept = hat.thresh, linetype='dashed', color='black')

    weight.plot = ggplot2::ggplot(cheungviechtdata, aes(y = weight, x = study, color = is.infl, group =
        geom_line(color = "black") + geom_point(size = 2) + scale_color_manual(values = c("blue", "red")
        theme_minimal() + theme(axis.title.x = element_blank(), legend.position = "none", axis.text.x =
        size = 5), axis.title.y = element_text(size = 7), axis.text.y = element_text(size = 5)) + ylab(
        scale_y_continuous(labels = scalefun)

    rma.influence.plot = arrangeGrob(rstudent.plot, dffits.plot, cook.d.plot, cov.r.plot, tau2.del.plot
        hat.plot, weight.plot, ncol = 2)

    # Perform Influence Analysis on meta object, generate forests
    meta.inf = meta::metainf(x, pooled = method.meta)

    if (x$sm %in% c("RR", "OR", "IRR")) {

        effect = x$sm
        n.studies = n.studies

        # Create Sortdat data set for sorting
```

```
sortdat = data.frame(studlab = meta.inf$studlab, mean = exp(meta.inf$TE), lower = exp(meta.inf$
    upper = exp(meta.inf$upper), i2 = meta.inf$I2)
sortdat2 = sortdat[1:(nrow(sortdat) - 2), ]
lastline = sortdat[nrow(sortdat), ]

# Change summary label
if (random == TRUE) {
    lastline[1] = "Random-Effects Model"
} else {
    lastline[1] = "Fixed-Effect Model"
}

for (i in 2:4) {
    lastline[i] = format(round(lastline[i], 2), nsmall = 2)
}

# Sort
sortdat.es = sortdat2[order(sortdat2$mean), ]
sortdat.es$studlab = factor(sortdat.es$studlab,
                            levels = sortdat.es$studlab[order(-sortdat.es$mean)])
sortdat.i2 = sortdat2[order(sortdat2$i2), ]
sortdat.i2$studlab = factor(sortdat.i2$studlab,
                            levels = sortdat.i2$studlab[order(-sortdat.i2$i2)])

# Generate Forest Plots
if (forest.lims[1] == "default") {

  if (min(sortdat.es$lower) > 0.5){
    min = 0.5
  } else {
    min = NA
  }

  if (max(sortdat.es$upper) <= 1){
    max = 1.2
  } else {
    max = round(max(sortdat.es$upper) + 6, 0)
  }

} else {

  if (forest.lims[1] <= 0){
    min = NA
  } else {
    min = forest.lims[1]
  }
    max = forest.lims[2] + 4
}

if (method.meta == "fixed"){
  plot.sum.effect = exp(x$TE.fixed)
  plot.sum.lower =  exp(x$lower.fixed)
  plot.sum.upper =  exp(x$upper.fixed)
```

```r
    } else {
      plot.sum.effect = exp(x$TE.random)
      plot.sum.lower =  exp(x$lower.random)
      plot.sum.upper =  exp(x$upper.random)
    }

    ########################################
    cat("===============")
    ########################################

    title.es = with(sortdat.es, {
      paste0("hat(theta)['*']~'='~", paste0("'", format(round(mean, 2)), "'"),
             "~'['*", paste0("'", format(round(lower, 2), nsmall = 2), "'"), "*'-'*",
             paste0("'", format(round(upper, 2), nsmall = 2), "'"), "*']'*';'~italic(I)^2~'='~",
             paste0("'", format(round(i2, 2)*100,nsmall = 0), "'"), "*'%'")})

    title.i2 = with(sortdat.i2,{
      paste0("italic(I)^2~'='~",
             paste0("'", format(round(i2, 2)*100,nsmall = 0), "'"), "*'%'", "*';'~",
             "hat(theta)['*']~'='~", paste0("'", format(round(mean, 2)), "'"),
             "~'['*", paste0("'", format(round(lower, 2), nsmall = 2), "'"), "*'-'*",
             paste0("'", format(round(upper, 2), nsmall = 2), "'"), "*']'")})

    forest.es = ggplot(sortdat.es, aes(x = studlab, y = mean, ymin = lower, ymax = upper)) +
        geom_text(aes(label = title.es, y = Inf), parse = T, hjust = "inward", size = 3 * text.scale
        color = "blue") + ylab(paste(effect, " (", as.character(lastline$studlab), ")", sep = "")) +
        coord_flip() + theme_minimal() + theme(axis.title.y = element_blank(), axis.title.x = elemen
        size = 12, face = "bold"), axis.text.y = element_text(color = "black", size = 9 * text.scale
        plot.title = element_text(face = "bold", hjust = 0.5), axis.line.x = element_line(color = "
        axis.ticks.x = element_line(color = "black"), axis.text.x = element_text(color = "black", si
            text.scale)) + scale_y_continuous(trans = "log2", limits = c(min, max)) +
      geom_rect(aes(ymin=plot.sum.lower, ymax=plot.sum.upper, xmin=0, xmax=Inf), alpha=0.08, fill="
      geom_hline(yintercept = plot.sum.effect, color = "darkgreen", linetype="dotted", size=0.5) +
      geom_point(shape = 15, size = 4.5, color = "grey") +
      geom_linerange(size = 0.9) +
      geom_pointrange(shape = 3, size = 0.3)

    forest.i2 = ggplot(sortdat.i2, aes(x = studlab, y = mean, ymin = lower, ymax = upper)) +
        geom_text(aes(label = title.i2, y = Inf), parse = T, hjust = "inward", size = 3 * text.scale
        color = "blue") + ylab(paste(effect, " (", as.character(lastline$studlab), ")", sep = "")) +
        coord_flip() + theme_minimal() + theme(axis.title.y = element_blank(), axis.title.x = elemen
        size = 12, face = "bold"), axis.text.y = element_text(color = "black", size = 9 * text.scale
        plot.title = element_text(face = "bold", hjust = 0.5), axis.line.x = element_line(color = "
        axis.ticks.x = element_line(color = "black"), axis.text.x = element_text(color = "black", si
            text.scale)) + scale_y_continuous(trans = "log2", limits = c(min, max)) +
      geom_rect(aes(ymin=plot.sum.lower, ymax=plot.sum.upper, xmin=0, xmax=Inf), alpha=0.08, fill="
      geom_hline(yintercept = plot.sum.effect, color = "darkgreen", linetype="dotted", size=0.5) +
      geom_point(shape = 15, size = 4.5, color = "grey") +
      geom_linerange(size = 0.9) +
      geom_pointrange(shape = 3, size = 0.3)

  } else if (class(x)[1] %in% c("metacor", "metaprop", "metarate")) {
```

```r
effect = x$sm
n.studies = n.studies

# Create Sortdat data set for sorting
sortdat = data.frame(studlab = meta.inf$studlab, mean = meta.inf$TE, lower = meta.inf$lower,
                     upper = meta.inf$upper, i2 = meta.inf$I2)
sortdat2 = sortdat[1:(nrow(sortdat) - 2), ]
lastline = sortdat[nrow(sortdat), ]

# Change summary label
if (random == TRUE) {
  lastline[1] = "Random-Effects Model"
} else {
  lastline[1] = "Fixed-Effect Model"
}

for (i in 2:4) {
  lastline[i] = format(round(lastline[i], 2), nsmall = 2)
}

# Sort
sortdat.es = sortdat2[order(sortdat2$mean), ]
sortdat.es$studlab = factor(sortdat.es$studlab,
                            levels = sortdat.es$studlab[order(-sortdat.es$mean)])
sortdat.i2 = sortdat2[order(sortdat2$i2), ]
sortdat.i2$studlab = factor(sortdat.i2$studlab,
                            levels = sortdat.i2$studlab[order(-sortdat.i2$i2)])

# Backtransform
backtransformer = function(x, sm, n){

  # Define functions
  z2cor = function(x)
  {
    res <- (exp(2 * x) - 1)/(exp(2 * x) + 1)
    res
  }

  logit2p = function(x)
  {
    res <- 1/(1 + exp(-x))
    res
  }

  asin2p = function (x, n = NULL, value = "mean", warn = TRUE)
  {
    if (all(is.na(x)))
      return(x)
    if (is.null(n)) {
      minimum <- asin(sqrt(0))
      maximum <- asin(sqrt(1))
    }
    else {
```

```
      minimum <- 0.5 * (asin(sqrt(0/(n + 1))) + asin(sqrt((0 +
                                                         1)/(n + 1))))
      maximum <- 0.5 * (asin(sqrt(n/(n + 1))) + asin(sqrt((n +
                                                         1)/(n + 1))))
    }
    sel0 <- x < minimum
    sel1 <- x > maximum
    if (any(sel0, na.rm = TRUE)) {
      if (is.null(n)) {
        if (warn)
          warning("Negative value for ", if (length(x) >
                                              1)
            "at least one ", if (value == "mean")
              "transformed proportion using arcsine transformation.\n  Proportion set to 0.",
            if (value == "lower")
              "lower confidence limit using arcsine transformation.\n  Lower confidence limit set
            if (value == "upper")
              "upper confidence limit using arcsine transformation.\n  Upper confidence limit set
            sep = "")
      }
      else {
        if (warn)
          warning("Too small value for ", if (length(x) >
                                               1)
            "at least one ", if (value == "mean")
              "transformed proportion using Freeman-Tukey double arcsine transformation.\n  Propo
            if (value == "lower")
              "lower confidence limit using Freeman-Tukey double arcsine transformation.\n  Lower
            if (value == "upper")
              "upper confidence limit using Freeman-Tukey double arcsine transformation.\n  Upper
            sep = "")
      }
    }
    if (any(sel1, na.rm = TRUE)) {
      if (is.null(n)) {
        if (warn)
          warning("Too large value for ", if (length(x) >
                                               1)
            "at least one ", if (value == "mean")
              "transformed proportion using arcsine transformation.\n  Proportion set to 1.",
            if (value == "lower")
              "lower confidence limit using arcsine transformation.\n  Lower confidence limit set
            if (value == "upper")
              "upper confidence limit using arcsine transformation.\n  Upper confidence limit set
            sep = "")
      }
      else {
        if (warn)
          warning("Too large value for ", if (length(x) >
                                               1)
            "at least one ", if (value == "mean")
              "transformed proportion using Freeman-Tukey double arcsine transformation.\n  Propo
            if (value == "lower")
```

```
            "lower confidence limit using Freeman-Tukey double arcsine transformation.\n  Lower
          if (value == "upper")
            "upper confidence limit using Freeman-Tukey double arcsine transformation.\n  Upper
          sep = "")
      }
    }
    res <- rep(NA, length(x))
    sel <- !(sel0 | sel1)
    sel <- !is.na(sel) & sel
    res[sel0] <- 0
    res[sel1] <- 1
    if (is.null(n)) {
      res[sel] <- sin(x[sel])^2
    }
    else {
      res[sel] <- 0.5 * (1 - sign(cos(2 * x[sel])) * sqrt(1 -
                                                           (sin(2 * x[sel]) + (sin(2 * x[sel]) -
    }
    res
  }

  asin2ir = function (x, time = NULL, value = "mean", warn = TRUE)
  {
    if (all(is.na(x)))
      return(x)
    minimum <- 0.5 * (sqrt(0/time) + sqrt((0 + 1)/time))
    sel0 <- x < minimum
    if (any(sel0, na.rm = TRUE)) {
      if (warn)
        warning("Too small value for ", if (length(x) > 1)
          "at least one ", if (value == "mean")
            "transformed proportion using Freeman-Tukey double arcsine transformation.\n  Rate se
          if (value == "lower")
            "lower confidence limit using Freeman-Tukey double arcsine transformation.\n  Lower c
          if (value == "upper")
            "upper confidence limit using Freeman-Tukey double arcsine transformation.\n  Upper c
          sep = "")
    }
    res <- rep(NA, length(x))
    sel <- !sel0
    sel <- !is.na(sel) & sel
    res[sel0] <- 0
    res[sel] <- (1/time[sel] - 8 * x[sel]^2 + 16 * time[sel] *
                   x[sel]^4)/(16 * x[sel]^2 * time[sel])
    res[res < 0] <- 0
    res
  }

  if(sm == "COR"){
    res = x
  }

  if(sm == "IR"){
```

```
      res = x
    }

    if(sm == "PRAW"){
      res = x
    }

    if(sm == "ZCOR"){
      res = z2cor(x)
    }

    if(sm == "PLOGIT"){
      res = logit2p(x)
    }

    if (sm == "PAS"){
      res <- asin2p(x, value = value, warn = FALSE)
    }

    if (sm == "PFT"){
      res = asin2p(x, n, value = value, warn = FALSE)
    }

    if (sm == "IRS"){
      res = x^2
    }

    if (sm == "IRFT"){
      res = asin2ir(x, time=n, value = value, warn = FALSE)
    }

    if (sm == "IRLN"){
      res = exp(x)
    }

    if (sm == "PLN"){
      res = exp(x)
    }

    res

  }

  if (class(x)[1] %in% c("metaprop", "metacor")){
    n.h.m = meta.inf$n.harmonic.mean[1:(length(meta.inf$n.harmonic.mean)-2)]
    n.h.m.tot = meta.inf$n.harmonic.mean[length(meta.inf$n.harmonic.mean)]
    n.h.m.es = n.h.m[order(sortdat.es$mean)]
    n.h.m.i2 = n.h.m[order(sortdat.i2$mean)]
    sortdat.es$mean = backtransformer(sortdat.es$mean, sm=effect, n=n.h.m.es)
    sortdat.es$lower = backtransformer(sortdat.es$lower, sm=effect, n=n.h.m.es)
    sortdat.es$upper = backtransformer(sortdat.es$upper, sm=effect, n=n.h.m.es)
    sortdat.i2$mean = backtransformer(sortdat.i2$mean, sm=effect, n=n.h.m.i2)
    sortdat.i2$lower = backtransformer(sortdat.i2$lower, sm=effect, n=n.h.m.i2)
    sortdat.i2$upper = backtransformer(sortdat.i2$upper, sm=effect, n=n.h.m.i2)
```

```r
    if (method.meta == "fixed"){
      plot.sum.effect = backtransformer(x$TE.fixed, sm=effect, n=n.h.m.tot)
      plot.sum.lower = backtransformer(x$lower.fixed, sm=effect, n=n.h.m.tot)
      plot.sum.upper = backtransformer(x$upper.fixed, sm=effect, n=n.h.m.tot)
    } else {
      plot.sum.effect = backtransformer(x$TE.random, sm=effect, n=n.h.m.tot)
      plot.sum.lower = backtransformer(x$lower.random, sm=effect, n=n.h.m.tot)
      plot.sum.upper = backtransformer(x$upper.random, sm=effect, n=n.h.m.tot)
    }

  } else {

    if(meta.inf$sm == "IRFT"){
      n.h.m = meta.inf$t.harmonic.mean[1:(length(meta.inf$t.harmonic.mean)-2)]
      n.h.m.es = n.h.m[order(sortdat.es$mean)]
      n.h.m.i2 = n.h.m[order(sortdat.i2$mean)]
      n.h.m.tot = meta.inf$t.harmonic.mean[length(meta.inf$t.harmonic.mean)]
      sortdat.es$mean = backtransformer(sortdat.es$mean, sm=effect, n=n.h.m.es)
      sortdat.es$lower = backtransformer(sortdat.es$lower, sm=effect, n=n.h.m.es)
      sortdat.es$upper = backtransformer(sortdat.es$upper, sm=effect, n=n.h.m.es)
      sortdat.i2$mean = backtransformer(sortdat.i2$mean, sm=effect, n=n.h.m.i2)
      sortdat.i2$lower = backtransformer(sortdat.i2$lower, sm=effect, n=n.h.m.i2)
      sortdat.i2$upper = backtransformer(sortdat.i2$upper, sm=effect, n=n.h.m.i2)

      if (method.meta == "fixed"){
        plot.sum.effect = backtransformer(x$TE.fixed, sm=effect, n=n.h.m.tot)
        plot.sum.lower = backtransformer(x$lower.fixed, sm=effect, n=n.h.m.tot)
        plot.sum.upper = backtransformer(x$upper.fixed, sm=effect, n=n.h.m.tot)
      } else {
        plot.sum.effect = backtransformer(x$TE.random, sm=effect, n=n.h.m.tot)
        plot.sum.lower = backtransformer(x$lower.random, sm=effect, n=n.h.m.tot)
        plot.sum.upper = backtransformer(x$upper.random, sm=effect, n=n.h.m.tot)
      }

    } else {
      n.h.m.tot = meta.inf$n.harmonic.mean[length(meta.inf$n.harmonic.mean)]
      sortdat.es$mean = backtransformer(sortdat.es$mean, sm=effect, n=NULL)
      sortdat.es$lower = backtransformer(sortdat.es$lower, sm=effect, n=NULL)
      sortdat.es$upper = backtransformer(sortdat.es$upper, sm=effect, n=NULL)
      sortdat.i2$mean = backtransformer(sortdat.i2$mean, sm=effect, n=NULL)
      sortdat.i2$lower = backtransformer(sortdat.i2$lower, sm=effect, n=NULL)
      sortdat.i2$upper = backtransformer(sortdat.i2$upper, sm=effect, n=NULL)

      if (method.meta == "fixed"){
        plot.sum.effect = backtransformer(x$TE.fixed, sm=effect, n=n.h.m.tot)
        plot.sum.lower = backtransformer(x$lower.fixed, sm=effect, n=n.h.m.tot)
        plot.sum.upper = backtransformer(x$upper.fixed, sm=effect, n=n.h.m.tot)
      } else {
        plot.sum.effect = backtransformer(x$TE.random, sm=effect, n=n.h.m.tot)
        plot.sum.lower = backtransformer(x$lower.random, sm=effect, n=n.h.m.tot)
        plot.sum.upper = backtransformer(x$upper.random, sm=effect, n=n.h.m.tot)
      }
```

```
    }
  }

  # Generate Forest Plots
  if (forest.lims[1] == "default") {
    if (class(x)[1] == "metacor"){
      min = min(sortdat.es$mean)-0.2
    } else {
      min = -0.2
    }

    max = max(sortdat.es$mean) + 0.5

  } else {
    min = forest.lims[1]
    max = forest.lims[2]
  }

  # Set ggtitles
  if (class(x)[1] == "metaprop"){
    ggtitl = as.character("Proportion")
  } else if (class(x)[1] == "metacor"){
    ggtitl = as.character("Correlation")
  } else {
    ggtitl = as.character("Rate")
  }

  ########################################
  cat("===============")
  ########################################

  title.es = with(sortdat.es, {
    paste0("hat(theta)['*']~'='~", paste0("'", format(round(mean, 2)), "'"),
           "~'['*", paste0("'", format(round(lower, 2), nsmall = 2), "'"), "*'-'*",
           paste0("'", format(round(upper, 2), nsmall = 2), "'"), "*']'*';'~italic(I)^2~'='~",
           paste0("'", format(round(i2, 2)*100,nsmall = 0), "'"), "*'%'")})

  title.i2 = with(sortdat.i2,{
    paste0("italic(I)^2~'='~",
           paste0("'", format(round(i2, 2)*100,nsmall = 0), "'"), "*'%'", "*';'~",
           "hat(theta)['*']~'='~", paste0("'", format(round(mean, 2)), "'"),
           "~'['*", paste0("'", format(round(lower, 2), nsmall = 2), "'"), "*'-'*",
           paste0("'", format(round(upper, 2), nsmall = 2), "'"), "*']'")})

  forest.es = ggplot(sortdat.es, aes(x = studlab, y = mean, ymin = lower, ymax = upper)) +
    geom_text(aes(label = title.es, y = Inf), parse = T, hjust = "inward", size = 3 * text.scale) +
    geom_hline(yintercept = 0, color = "blue") + ylab(paste(ggtitl, " (", as.character(lastline$stu
    ggtitle(paste("Sorted by", ggtitl)) +
    coord_flip() +
    theme_minimal() +
    theme(axis.title.y = element_blank(), axis.title.x = element_text(color = "black", size = 12, fa
    scale_y_continuous(limits = c(min, max)) +
    geom_rect(aes(ymin=plot.sum.lower, ymax=plot.sum.upper, xmin=0, xmax=Inf), alpha=0.08, fill="lig
```

```r
    geom_hline(yintercept = plot.sum.effect, color = "darkgreen", linetype="dotted", size=0.5) +
    geom_point(shape = 15, size = 4.5, color = "grey") +
    geom_linerange(size = 0.9) +
    geom_pointrange(shape = 3, size = 0.3)

  forest.i2 = ggplot(sortdat.i2, aes(x = studlab, y = mean, ymin = lower, ymax = upper)) +
    geom_text(aes(label = title.i2, y = Inf), parse = T, hjust = "inward", size = 3 * text.scale) +
    geom_hline(yintercept = 0, color = "blue") + ylab(paste(ggtitl, " (", as.character(lastline$stud
    ggtitle(expression(Sorted~by~italic(I)^2)) +
    coord_flip() +
    theme_minimal() +
    theme(axis.title.y = element_blank(), axis.title.x = element_text(color = "black", size = 12, fa
    scale_y_continuous(limits = c(min, max)) +
    geom_rect(aes(ymin=plot.sum.lower, ymax=plot.sum.upper, xmin=0, xmax=Inf), alpha=0.08, fill="lig
    geom_hline(yintercept = plot.sum.effect, color = "darkgreen", linetype="dotted", size=0.5) +
    geom_point(shape = 15, size = 4.5, color = "grey") +
    geom_linerange(size = 0.9) +
    geom_pointrange(shape = 3, size = 0.3)

} else {

    # Create Sortdat data set for sorting
    sortdat = data.frame(studlab = meta.inf$studlab, mean = meta.inf$TE, lower = meta.inf$lower, upp
        i2 = meta.inf$I2)
    sortdat2 = sortdat[1:(nrow(sortdat) - 2), ]
    lastline = sortdat[nrow(sortdat), ]

    # Change summary label
    if (random == TRUE) {
        lastline[1] = "Random-Effects Model"
    } else {
        lastline[1] = "Fixed-Effect Model"
    }

    for (i in 2:4) {
        lastline[i] = format(round(lastline[i], 2), nsmall = 2)
    }

    # Sort
    sortdat.es = sortdat2[order(sortdat2$mean), ]
    sortdat.es$studlab = factor(sortdat.es$studlab,
                                levels = sortdat.es$studlab[order(-sortdat.es$mean)])
    sortdat.i2 = sortdat2[order(sortdat2$i2), ]
    sortdat.i2$studlab = factor(sortdat.i2$studlab,
                                levels = sortdat.i2$studlab[order(-sortdat.i2$i2)])

    # Generate Forest Plots
    if (forest.lims[1] == "default") {
        min = round(min(sortdat.es$lower) - 0.1, 2)
        max = round(max(sortdat.es$upper) + 0.3, 2)
    } else {
        min = forest.lims[1]
        max = forest.lims[2]
```

```
    }

    if (method.meta == "fixed"){
      plot.sum.effect = x$TE.fixed
      plot.sum.lower = x$lower.fixed
      plot.sum.upper = x$upper.fixed
    } else {
      plot.sum.effect = x$TE.random
      plot.sum.lower = x$lower.random
      plot.sum.upper = x$upper.random
    }

    ########################################
    cat("===============")
    ########################################

    title.es = with(sortdat.es, {
                     paste0("hat(theta)['*']~'='~", paste0("'", format(round(mean, 2)), "'"),
                            "~'['*", paste0("'", format(round(lower, 2), nsmall = 2), "'"), "*'-'*"
                            paste0("'", format(round(upper, 2), nsmall = 2), "'"), "*']'*';'~italic
                            paste0("'", format(round(i2, 2)*100,nsmall = 0), "'"), "*'%'")})

    title.i2 = with(sortdat.i2,{
                      paste0("italic(I)^2~'='~",
                             paste0("'", format(round(i2, 2)*100,nsmall = 0), "'"), "*'%'", "*';'~"
                             "hat(theta)['*']~'='~", paste0("'", format(round(mean, 2)), "'"),
                             "~'['*", paste0("'", format(round(lower, 2), nsmall = 2), "'"), "*'-'*
                             paste0("'", format(round(upper, 2), nsmall = 2), "'"), "*']'")})

    forest.es = ggplot(sortdat.es, aes(x = studlab, y = mean, ymin = lower, ymax = upper)) +
        geom_text(aes(label = title.es, y = Inf), parse = T, hjust = "inward", size = 3 * text.scal
        color = "blue") + ylim(min, max) + ylab(paste("Effect Size (", as.character(lastline$studla
        ")", sep = "")) + ggtitle("Sorted by Effect Size") + coord_flip() + theme_minimal() + theme
        axis.title.x = element_text(color = "black", size = 12, face = "bold"), axis.text.y = eleme
            size = 9 * text.scale), plot.title = element_text(face = "bold", hjust = 0.5), axis.lin
        axis.ticks.x = element_line(color = "black"), axis.text.x = element_text(color = "black", s
            text.scale)) +
      geom_rect(aes(ymin=plot.sum.lower, ymax=plot.sum.upper, xmin=0, xmax=Inf), alpha=0.08, fill="
      geom_hline(yintercept = plot.sum.effect, color = "darkgreen", linetype="dotted", size=0.5) +
      geom_point(shape = 15, size = 4.5, color = "grey") +
      geom_linerange(size = 0.9) +
      geom_pointrange(shape = 3, size = 0.3)

    forest.i2 = ggplot(sortdat.i2, aes(x = studlab, y = mean, ymin = lower, ymax = upper)) +
      geom_text(aes(label = title.i2, y = Inf), parse = T, hjust = "inward", size = 3 * text.scale)
      color = "blue") + ylim(min, max) + ylab(paste("Effect Size (", as.character(lastline$studlab)
      ggtitle(expression(Sorted~by~italic(I)^2)) + coord_flip() + theme_minimal() + theme(axis.titl
      axis.title.x = element_text(color = "black", size = 12, face = "bold"),
      axis.text.y = element_text(color = "black",size = 9 * text.scale), plot.title = element_text(
      axis.line.x = element_line(color = "black"),
      axis.ticks.x = element_line(color = "black"), axis.text.x = element_text(color = "black", siz
      geom_rect(aes(ymin=plot.sum.lower, ymax=plot.sum.upper, xmin=0, xmax=Inf), alpha=0.08, fill="
      geom_hline(yintercept = plot.sum.effect, color = "darkgreen", linetype="dotted", size=0.5) +
```

```
        geom_point(shape = 15, size = 4.5, color = "grey") +
        geom_linerange(size = 0.9) +
        geom_pointrange(shape = 3, size = 0.3)
}

# Generate baujat plot Define baujat.silent
baujat.silent = function(x, yscale = 1, xlim, ylim, ...) {

    TE = x$TE
    seTE = x$seTE
    TE.fixed = metagen(TE, seTE, exclude = x$exclude)$TE.fixed
    k = x$k
    studlab = x$studlab
    SE = x$seTE

    m.inf = metainf(x, pooled = "fixed")
    TE.inf = m.inf$TE[1:length(TE)]
    seTE.inf = m.inf$seTE[1:length(TE)]

    ys = (TE.inf - TE.fixed)^2/seTE.inf^2
    ys = ys * yscale

    xs = (TE - TE.fixed)^2/seTE^2

    if (!is.null(x$exclude))
        xs[x$exclude] = 0

    res = data.frame(studlab = studlab, x = xs, y = ys, se = SE)

    return(res)
}

########################################
cat("===============")
########################################

bjt = baujat.silent(x)

BaujatPlot = ggplot(bjt, aes(x = x, y = y)) + geom_point(aes(size = (1/se)), color = "blue", alpha =
    geom_rug(color = "lightgray", alpha = 0.5) + theme(legend.position = "none") + xlab("Overall het
    ylab("Influence on pooled result") + geom_label_repel(label = bjt$studlab, color = "black", size
    text.scale, alpha = 0.7)

# Return

########################################
cat("===============] DONE \n")
########################################

# Prepare data for return
return.data = cbind(sortdat2, cheungviechtdata[, 2:ncol(cheungviechtdata)], HetContrib = bjt$x, Infl

if (x$sm %in% c("RR", "OR", "IRR")) {colnames(return.data)[1:2] = c("Author", effect)}
```

```r
    else {colnames(return.data)[1:2] = c("Author", "effect")}

    returnlist = suppressWarnings(suppressMessages(list(BaujatPlot = BaujatPlot,
                     InfluenceCharacteristics = rma.influence.plot,
                     ForestEffectSize = forest.es,
                     ForestI2 = forest.i2,
                     Data = return.data)))

    if (return.separate.plots == T){class(returnlist) = c("InfluenceAnalysis", "rsp")}
    if (return.separate.plots == F){class(returnlist) = c("InfluenceAnalysis", "rsp.null")}

    returnlist

}
```

```r
# Returns an error
meta1_inf <- InfluenceAnalysis(meta1, random = TRUE)
```

GOSH plot analysis

```
library(metafor)
```

```
## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack

##
## Loading the 'metafor' package (version 3.0-2). For an
## introduction to the package please type: help(metafor)
```

```
### calculate r-to-z transformed correlations and corresponding sampling variances
df3 <- escalc(measure="ZCOR", ri=r, ni=n, data=df2)

### meta-analysis of the transformed correlations using a random-effects model
meta1_rma <- rma(yi = yi,
                 vi = vi,
                 data = df3)
meta1_rma
```

```
##
## Random-Effects Model (k = 273; tau^2 estimator: REML)
##
## tau^2 (estimated amount of total heterogeneity): 0.0187 (SE = 0.0023)
## tau (square root of estimated tau^2 value):      0.1369
## I^2 (total heterogeneity / total variability):   95.68%
## H^2 (total variability / sampling variability):  23.13
##
## Test for Heterogeneity:
## Q(df = 272) = 8958.2395, p-val < .0001
##
## Model Results:
##
## estimate      se     zval    pval   ci.lb   ci.ub
##   0.1071  0.0103  10.4110  <.0001  0.0869  0.1272  ***
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

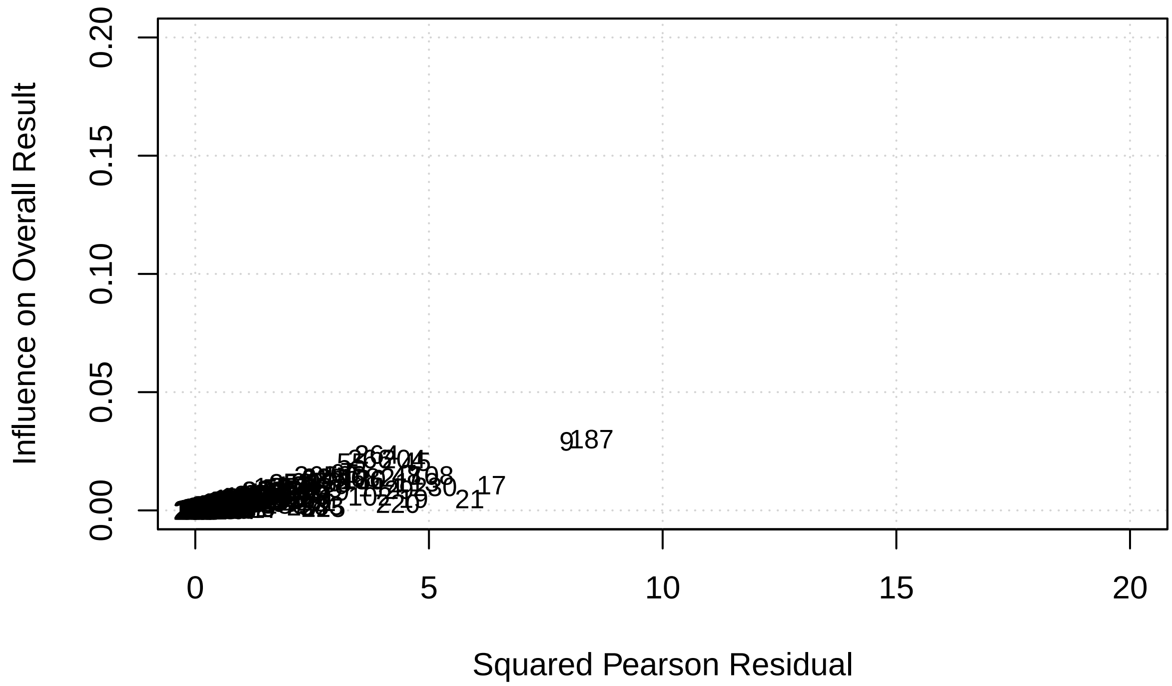
```
### average correlation with 95% CI
predict(meta1_rma, transf=transf.ztor)
```

```
##
##    pred  ci.lb  ci.ub   pi.lb  pi.ub
##  0.1067 0.0867 0.1265 -0.1606 0.3594
```

```
# GOSH analysis  (time: ~ 1 day and 5 hours)
meta1_gosh <- gosh(meta1_rma)
```

Other ways to do outlier and influence analysis

```
# Baujat plot
# https://www.metafor-project.org/doku.php/plots:baujat_plot
baujat(meta1_rma, xlim=c(0,20), ylim=c(0,0.2))
```




```
# https://www.dsquintana.blog/how-do-you-decide-which-studies-in-a-meta-analysis-are-influential-and-sh
# https://www.metafor-project.org/doku.php/plots:plot_of_influence_diagnostics
inf <- influence(meta1_rma)
plot(inf, layout=c(8,1))
```

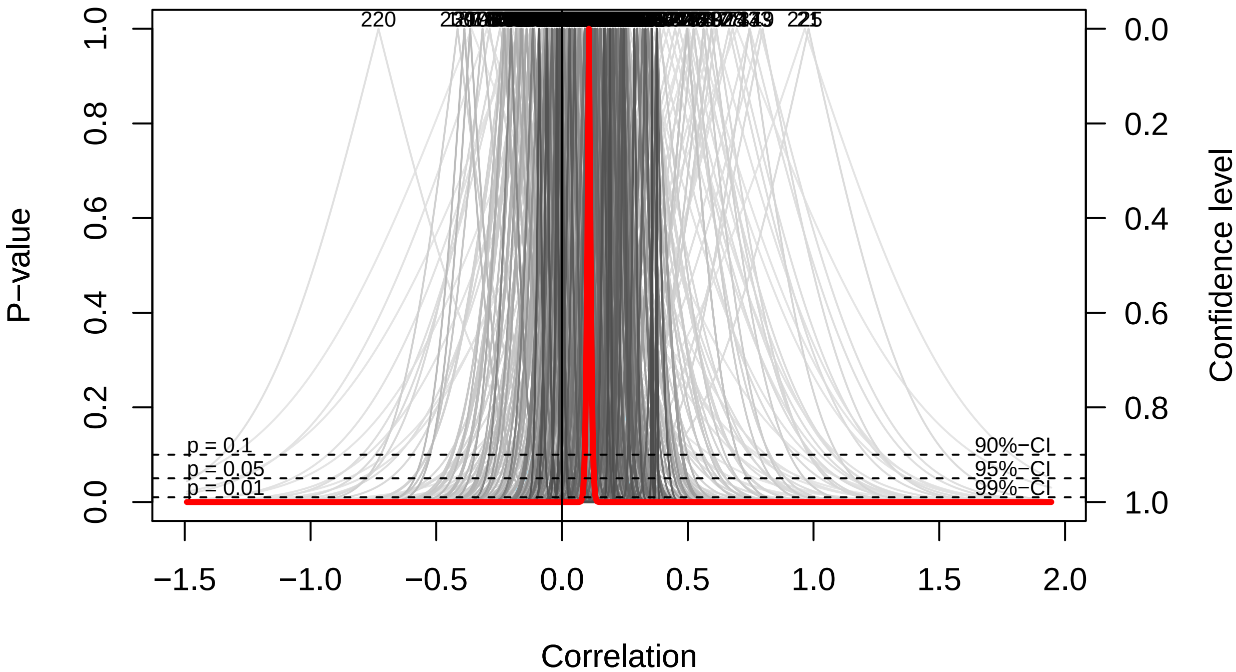
Forest plot

```
pdf(file='forestplot.pdf', width = 10, height = 60)
forest.meta(meta1,
            sortvar = r)
dev.off()
```

Drapery plots

Weird, not sure if it is using the z-transformed or if this plot does not work well with correlations. Values are over-bounds.

```
drapery(meta1,
        type = 'pval',
        legend = FALSE)
```



Subgroup analysis

Sisks et al. (2018) found as **significant** moderators:

- Student factors:
  - Developmental stage of the student: children, adolescents, adults
- Developmental stage as a moderator of mind-set on GPA

Sisks et al. (2018) found as **non- significant** moderators:

- Student factors:
  - Academic risk status: low-risk, moderately at risk, highly at-risk students
  - Socioeconomic status
- Academic achievement measure: Course exam, Course grade, GPA, Standardized test

```
# Significant
levels(df2$development_stage)
```

```
## [1] "Wide Range"  "Adolescents" "Adults"      "Children"
```

```
# Non-significants
levels(df2$risk_status)
```

```
## [1] "."         "high"      "low"       "moderate"
```

```
levels(df2$ses)
```

```
## [1] "low SES"      "not low"      "not reported"
```

```
levels(df2$academic_achievement_measure_type)
```

```
## [1] "Course exam"       "Course grade"      "GPA"
## [4] "Standardized test"
```

Developmental stage

```
levels(df2$development_stage)
```

```
## [1] "Wide Range"  "Adolescents" "Adults"      "Children"
```

```
# Subset to include only studies with homogeneous groups
# meaning: exclude "Wide range" and "Wide Range"
df2_develop <- filter(df2,
                      (development_stage == "Adolescents") |
                        (development_stage == "Adults") |
                        (development_stage == "Children"))

df2_develop$development_stage <- droplevels(df2_develop$development_stage)
levels(df2_develop$development_stage)
```

```
## [1] "Adolescents" "Adults"      "Children"
```

```
# Fit meta-analysis: DIFFERENT TAU^2
meta1_develop1 <- metacor(cor = r,
                         n = n,
                         studlab = reference,
                         data= df2_develop,
                         fixed = FALSE,
                         random = TRUE,
                         method.tau = 'REML',
```

```r
                         hakn = FALSE,
                         title = "Mindset and Academic Achievement",
                         prediction = TRUE,
                         subgroup = development_stage,
                         tau.common = FALSE)
meta1_develop1
```

```
## Review:     Mindset and Academic Achievement
##
## Number of studies combined: k = 265
## Number of observations: o = 413721
##
##                            COR            95%-CI     z  p-value
## Random effects model 0.1064 [ 0.0858; 0.1270] 10.04 < 0.0001
## Prediction interval         [-0.1669; 0.3646]
##
## Quantifying heterogeneity:
##  tau^2 = 0.0194 [0.0168; 0.0294]; tau = 0.1394 [0.1297; 0.1716]
##  I^2 = 97.0% [96.8%; 97.2%]; H = 5.78 [5.60; 5.97]
##
## Test of heterogeneity:
##        Q d.f. p-value
##  8829.06  264       0
##
## Results for subgroups (random effects model):
##                                      k    COR            95%-CI  tau^2    tau
## development_stage = Adolescents 126 0.1490 [ 0.1185; 0.1793] 0.0192 0.1386
## development_stage = Adults       89 0.0212 [-0.0054; 0.0476] 0.0108 0.1040
## development_stage = Children     50 0.1955 [ 0.1573; 0.2330] 0.0081 0.0899
##                                          Q   I^2
## development_stage = Adolescents 4868.99 97.4%
## development_stage = Adults       394.68 77.7%
## development_stage = Children     131.74 62.8%
##
## Test for subgroup differences (random effects model):
##                        Q d.f.  p-value
## Between groups   67.79    2 < 0.0001
##
## Details on meta-analytical method:
## - Inverse variance method
## - Restricted maximum-likelihood estimator for tau^2
## - Q-profile method for confidence interval of tau^2 and tau
## - Fisher's z transformation of correlations
```

```r
# Fit meta-analysis: EQUAL TAU^2
meta1_develop2 <- metacor(cor = r,
                         n = n,
                         studlab = reference,
                         data= df2_develop,
                         fixed = FALSE,
                         random = TRUE,
                         method.tau = 'REML',
                         hakn = FALSE,
                         title = "Mindset and Academic Achievement",
```

```r
                         prediction = TRUE,
                         subgroup = development_stage,
                         tau.common = TRUE)
meta1_develop2
```

```
## Review:     Mindset and Academic Achievement
##
## Number of studies combined: k = 265
## Number of observations: o = 413721
##
##                            COR            95%-CI     z  p-value
## Random effects model 0.1064 [ 0.0858; 0.1270] 10.04 < 0.0001
## Prediction interval         [-0.1669; 0.3646]
##
## Quantifying heterogeneity:
##  tau^2 = 0.0194 [0.0168; 0.0294]; tau = 0.1394 [0.1297; 0.1716]
##  I^2 = 97.0% [96.8%; 97.2%]; H = 5.78 [5.60; 5.97]
##
## Quantifying residual heterogeneity:
##  tau^2 = 0.0143; tau = 0.1194; I^2 = 95.1% [94.8%; 95.5%]; H = 4.54 [4.37; 4.71]
##
## Test of heterogeneity:
##        Q d.f. p-value
##  8829.06  264       0
##
## Results for subgroups (random effects model):
##                                    k    COR            95%-CI  tau^2    tau
## development_stage = Adolescents 126 0.1496 [ 0.1224; 0.1766] 0.0143 0.1194
## development_stage = Adults       89 0.0205 [-0.0088; 0.0498] 0.0143 0.1194
## development_stage = Children     50 0.1985 [ 0.1527; 0.2435] 0.0143 0.1194
##                                        Q   I^2
## development_stage = Adolescents 4868.99 97.4%
## development_stage = Adults       394.68 77.7%
## development_stage = Children     131.74 62.8%
##
## Test for subgroup differences (random effects model):
##                        Q d.f.  p-value
## Between groups   58.15    2 < 0.0001
## Within groups  5395.40  262        0
##
## Details on meta-analytical method:
## - Inverse variance method
## - Restricted maximum-likelihood estimator for tau^2
##   (assuming common tau^2 in subgroups)
## - Q-profile method for confidence interval of tau^2 and tau
## - Fisher's z transformation of correlations
```

```r
## TO BE DONE: HOW TO COMPARE THE THREE CATEGOREIS (ANOVA WITH CORRECTION FOR MULTIPLE COMPARISON)
```

Multilevel meta-analysis

```r
library(metafor)

meta1_multi <- rma.mv(yi = yi,
                      V = vi,
                      random = list(~1 | study_id/es_id),
                      data = df3)
summary(meta1_multi)
```

```
##
## Multivariate Meta-Analysis Model (k = 273; method: REML)
##
##    logLik   Deviance        AIC        BIC       AICc
##   83.4307  -166.8614  -160.8614  -150.0440  -160.7718
##
## Variance Components:
##
##             estim    sqrt  nlvls  fixed          factor
## sigma^2.1  0.0153  0.1238    129     no        study_id
## sigma^2.2  0.0041  0.0644    273     no  study_id/es_id
##
## Test for Heterogeneity:
## Q(df = 272) = 8958.2395, p-val < .0001
##
## Model Results:
##
## estimate      se    zval    pval   ci.lb   ci.ub
##   0.0906  0.0132  6.8419  <.0001  0.0647  0.1166  ***
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- Variance components:
  - The between-study variance is sigma^2.1 = 0.0153 (equivalent to tau^2 in REM)- At this level we have 129 studies.
  - The within-study variance is sigma^2.2 = 0.0041. At this level we have 273 effect sizes.
- Estimate: 0.0906. However, this is a Fisher's z, so we have to transform the effect back to a normal correlation:

```r
transf.ztor(meta1_multi$b)
```

```
##               [,1]
## intrcpt 0.09038732
```

- The correlation is approximately 0.09 which is a modest association
- The test for heterogeneity rejects the null hypothesis of homogeneity, however this is not informative. Instead, we should look at the amount of heterogeneity caputred by each level in our model.

Distribution of variance across levels

Function:

```
#' Calculate I-squared values and variance distribution for multilevel meta-analysis models
#'
#' This function calculates values of \eqn{I^2} and the variance distribution for multilevel meta-analys
#' models fitted with \code{\link[metafor]{rma.mv}}.
#'
#'
#' @usage mlm.variance.distribution(x)
#'
#' @param x An object of class \code{rma.mv}. Must be a multilevel model with two random effects (three
#'
#' @details This function estimates the distribution of variance in a three-level meta-analysis
#' model (fitted with the \code{\link[metafor]{rma.mv}} function). The share of variance attributable t
#' sampling error, within and between-cluster heterogeneity is calculated,
#' and an estimate of \eqn{I^2} (total and for Level 2 and Level 3) is provided. The function uses the
#' Cheung (2014) to estimate the variance proportions attributable to each model component and to deriv
#'
#'
#' @references
#'
#' Harrer, M., Cuijpers, P., Furukawa, T.A, & Ebert, D. D. (2019).
#' \emph{Doing Meta-Analysis in R: A Hands-on Guide}. DOI: 10.5281/zenodo.2551803. \href{https://bookdo
#'
#'Cheung, M. W. L. (2014). Modeling dependent effect sizes with three-level meta-analyses: a structural
#'
#' @author Mathias Harrer & David Daniel Ebert
#'
#' @aliases var.comp
#'
#' @import ggplot2
#' @importFrom stats model.matrix
#'
#' @return Returns a data frame containing the results. A plot summarizing the variance distribution an
#'
#' @export mlm.variance.distribution
#' @export var.comp
#'
#' @examples
#' # Use dat.konstantopoulos2011 from the "metafor" package
#' library(metafor)
#'
#' # Build Multilevel Model (Three Levels)
#' m = rma.mv(yi, vi, random = ~ 1 | district/school, data=dat.konstantopoulos2011)
#'
#' # Calculate Variance Distribution
#' mlm.variance.distribution(m)
#'
#' # Use alias 'var.comp' and 'Chernobyl' data set
#' data("Chernobyl")
#' m2 = rma.mv(yi = z, V = var.z, data = Chernobyl, random = ~ 1 | author/es.id)
#' res = var.comp(m2)
#'
```

```r
#' # Print results
#' res
#'
#' # Generate plot
#' plot(res)



mlm.variance.distribution = var.comp = function(x){

  m = x

  # Check class
  if (!(class(m)[1] %in% c("rma.mv", "rma"))){
    stop("x must be of class 'rma.mv'.")
  }

  # Check for three level model
  if (m$sigma2s != 2){
    stop("The model you provided does not seem to be a three-level model. This function can only be used
  }

  # Check for right specification (nested model)
  if (sum(grepl("/", as.character(m$random[[1]]))) < 1){
    stop("Model must contain nested random effects. Did you use the '~ 1 | cluster/effect-within-cluste
  }

  # Get variance diagonal and calculate total variance
  n = m$k.eff
  vector.inv.var = 1/(diag(m$V))
  sum.inv.var = sum(vector.inv.var)
  sum.sq.inv.var = (sum.inv.var)^2
  vector.inv.var.sq = 1/(diag(m$V)^2)
  sum.inv.var.sq = sum(vector.inv.var.sq)
  num = (n-1)*sum.inv.var
  den = sum.sq.inv.var - sum.inv.var.sq
  est.samp.var = num/den

  # Calculate variance proportions
  level1=((est.samp.var)/(m$sigma2[1]+m$sigma2[2]+est.samp.var)*100)
  level2=((m$sigma2[2])/(m$sigma2[1]+m$sigma2[2]+est.samp.var)*100)
  level3=((m$sigma2[1])/(m$sigma2[1]+m$sigma2[2]+est.samp.var)*100)

  # Prepare df for return
  Level=c("Level 1", "Level 2", "Level 3")
  Variance=c(level1, level2, level3)
  df.res=data.frame(Variance)
  colnames(df.res) = c("% of total variance")
  rownames(df.res) = Level
  I2 = c("---", round(Variance[2:3], 2))
  df.res = as.data.frame(cbind(df.res, I2))

  totalI2 = Variance[2] + Variance[3]
```

```r
# Generate plot
df1 = data.frame("Level" = c("Sampling Error", "Total Heterogeneity"),
                "Variance" = c(df.res[1,1], df.res[2,1]+df.res[3,1]),
                "Type" = rep(1,2))

df2 = data.frame("Level" = rownames(df.res),
                 "Variance" = df.res[,1],
                 "Type" = rep(2,3))

df = as.data.frame(rbind(df1, df2))


g = ggplot(df, aes(fill=Level, y=Variance, x=as.factor(Type))) +
  coord_cartesian(ylim = c(0,1), clip = "off") +
  geom_bar(stat="identity", position="fill", width = 1, color="black") +
  scale_y_continuous(labels = scales::percent)+
  theme(axis.title.x=element_blank(),
        axis.text.y = element_text(color="black"),
        axis.line.y = element_blank(),
        axis.title.y=element_blank(),
        axis.line.x = element_blank(),
        axis.ticks.x = element_blank(),
        axis.text.x = element_blank(),
        axis.ticks.y = element_line(lineend = "round"),
        legend.position = "none",
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.background = element_blank(),
        legend.background = element_rect(linetype="solid",
                                         colour ="black"),
        legend.title = element_blank(),
        legend.key.size = unit(0.75,"cm"),
        axis.ticks.length=unit(.25, "cm"),
        plot.margin = unit(c(1,3,1,1), "lines")) +
  scale_fill_manual(values = c("darkseagreen3", "deepskyblue3", "darkseagreen2",
                               "deepskyblue1", "deepskyblue2")) +

  # Add Annotation

  # Total Variance
  annotate("text", x = 1.5, y = 1.05,
           label = paste("Total Variance:",
                         round(m$sigma2[1]+m$sigma2[2]+est.samp.var, 3))) +

  # Sampling Error
  annotate("text", x = 1, y = (df[1,2]/2+df[2,2])/100,
           label = paste("Sampling Error Variance: \n", round(est.samp.var, 3)), size = 3) +

  # Total I2
  annotate("text", x = 1, y = ((df[2,2])/100)/2-0.02,
           label = bquote("Total"~italic(I)^2*":"~.(round(df[2,2],2))*"%"), size = 3) +
  annotate("text", x = 1, y = ((df[2,2])/100)/2+0.05,
```

```
             label = paste("Variance not attributable \n to sampling error: \n", round(m$sigma2[1]+m$si

    # Level 1
    annotate("text", x = 2, y = (df[1,2]/2+df[2,2])/100, label = paste("Level 1: \n",
                                                                     round(df$Variance[3],2), "%", se

    # Level 2
    annotate("text", x = 2, y = (df[5,2]+(df[4,2]/2))/100,
             label = bquote(italic(I)[Level2]^2*":"~.(round(df[4,2],2))*"%"), size = 3) +

    # Level 3
    annotate("text", x = 2, y = (df[5,2]/2)/100,
             label = bquote(italic(I)[Level3]^2*":"~.(round(df[5,2],2))*"%"), size = 3)

  returnlist = list(results = df.res,
                    totalI2 = totalI2,
                    plot = g)
  class(returnlist) = c("mlm.variance.distribution", "list")

  invisible(returnlist)

  returnlist

}
```

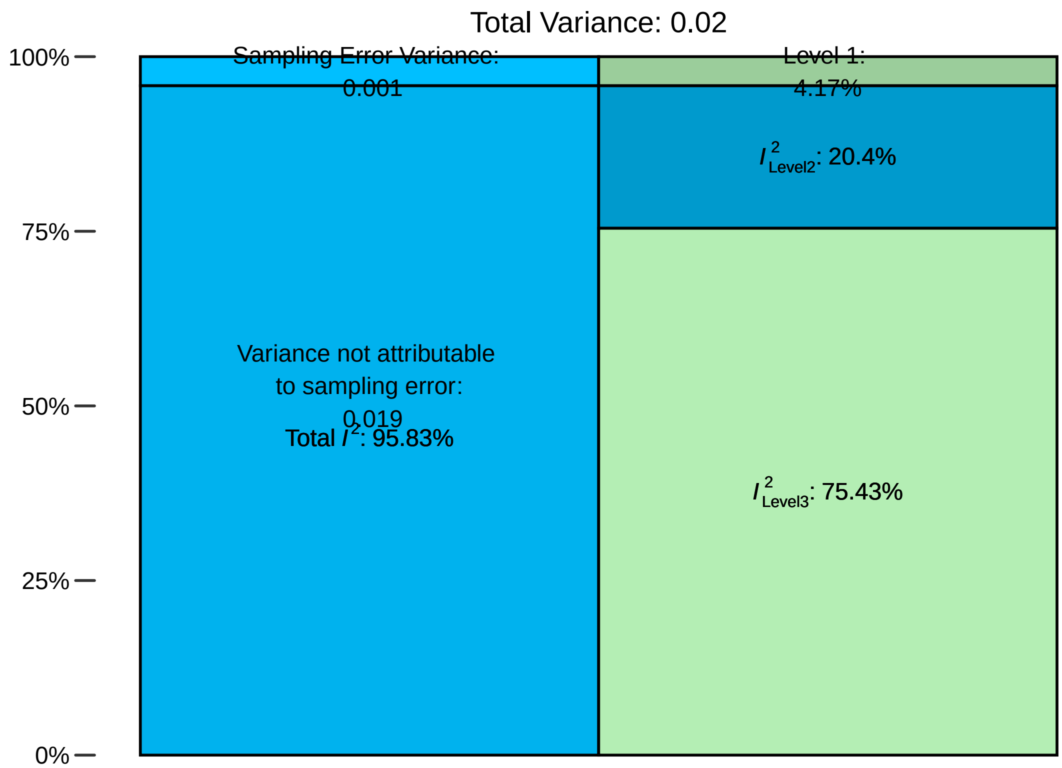
```
i_squared <- var.comp(meta1_multi)
i_squared
```

```
## $results
##         % of total variance    I2
## Level 1            4.167039   ---
## Level 2           20.402266  20.4
## Level 3           75.430695 75.43
##
## $totalI2
## [1] 95.83296
##
## $plot

## Warning in is.na(x): is.na() applied to non-(list or vector) of type 'language'

## Warning in is.na(x): is.na() applied to non-(list or vector) of type 'language'

## Warning in is.na(x): is.na() applied to non-(list or vector) of type 'language'
```




```
##
## attr(,"class")
## [1] "mlm.variance.distribution" "list"
```

- The sampling error variance at level 1 is small, making up about 4% of the total variance.
- Iˆ2 at level 2 is about 20%. This is the amount of heterogeneity variance within clusters.
- Iˆ2 at level 3 is about 75%. This is the amount of between-study heterogeneity.

Comparing models

```
meta1_2level <- rma.mv(yi = yi,
                       V = vi,
                       random = list(~1 | study_id/es_id),
                       data = df3,
                       sigma2 = c(0, NA))
summary(meta1_2level)
```

```
##
## Multivariate Meta-Analysis Model (k = 273; method: REML)
##
##    logLik   Deviance        AIC        BIC       AICc
##   60.3140  -120.6280  -116.6280  -109.4164  -116.5834
##
## Variance Components:
##
##              estim    sqrt  nlvls  fixed          factor
## sigma^2.1  0.0000  0.0000    129    yes        study_id
## sigma^2.2  0.0187  0.1369    273     no  study_id/es_id
##
## Test for Heterogeneity:
## Q(df = 272) = 8958.2395, p-val < .0001
##
## Model Results:
##
## estimate      se     zval    pval   ci.lb   ci.ub
##   0.1071  0.0103  10.4112  <.0001  0.0869  0.1272  ***
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(meta1_multi, meta1_2level)
```

```
##
##         df       AIC       BIC      AICc  logLik     LRT   pval        QE
## Full     3 -160.8614 -150.0440 -160.7718 83.4307                 8958.2395
## Reduced  2 -116.6280 -109.4164 -116.5834 60.3140 46.2334 <.0001 8958.2395
```

The three-level model has better fit: AIC and BIC are lower for the 3-level model and the LRT is significant

Subgroup analysis

```
# Check levels
levels(df3$development_stage)
```

Developmental stage

```
## [1] "Wide Range"  "Adolescents" "Adults"      "Children"
```

```
# Include only levels of interest
df3_develop <- filter(df3,
                      (development_stage == "Adolescents") |
                        (development_stage == "Adults") |
                        (development_stage == "Children"))
```

```r
levels(df3_develop$development_stage)
```

```
## [1] "Wide Range"  "Adolescents" "Adults"      "Children"
```

```r
# Drop level "Wide Range"
df3 %>% filter(development_stage == "Wide Range")
```

```
##   document_id study_id sample_id sample_country es_id
## 1          18       19        30            USA    56
## 2          39       42        55       Portugal    95
## 3          39       42        55       Portugal    96
## 4          52       56        78            USA   134
## 5          70       74        97         Greece   159
## 6          72       76        99            USA   161
## 7         112      118       151            USA   257
## 8         112      118       151            USA   258
##                        reference    n adjusted_n
## 1              Clevenger (2013)  190   190.0000
## 2   Faria & Fontaine (1997) - M1 1500   999.3333
## 3   Faria & Fontaine (1997) - M2 1500   999.3333
## 4          Haimovitz et al. (2011)  897   897.0000
## 5       Leondari & Gialamas (2002)  451   451.0000
## 6                 Lindsay (2006)   87    87.0000
## 7 Tucker-Drob et al. (2016) - M1  749   390.5648
## 8 Tucker-Drob et al. (2016) - M2  759   396.0207
##                                       student_description
## 1 4th-8th graders at a charter school in the southeast.
## 2                  junior high and senior high students
## 3                  junior high and senior high students
## 4     3rd-8th graders, 14-17% of students at each grade
## 5          elementary and junior high students in Greece
## 6              5th-8th graders from Arapahoe School in WY
## 7                              3rd- through 8th-graders
## 8                              3rd- through 8th-graders
##                school_level development_stage risk_status          ses
## 1           primary and middle        Wide Range         low      not low
## 2 elementary, middle and high        Wide Range         low      not low
## 3 elementary, middle and high        Wide Range         low      not low
## 4           primary and middle        Wide Range         low not reported
## 5           primary and middle        Wide Range         low      low SES
## 6           primary and middle        Wide Range         low not reported
## 7           primary and middle        Wide Range         low not reported
## 8           primary and middle        Wide Range         low not reported
##
## 1
## 2
## 3
## 4 Entity theory of intelligence (from Dweck, 1999; also used the Cain and Dweck 1995 version for 3rd
## 5
## 6                                                                                                 Impl
## 7
## 8
##
## 1                                                                            6 items, 3 entity, 3 incre
```

```
## 2 11 growth statements about dynamic (i.e., changeable/growth) intelligence, rated 1-6, included Dwe
## 3  15 fixed statements about static (i.e., fixed) intelligence rated on a scale of 1-6, included Dwe
## 4                                                                                 3 fixed theory 
## 5
## 6                                                                                              6 
## 7                                                                                              6
## 8                                                                                              6
##   mindset_type
## 1 Intelligence
## 2 Intelligence
## 3 Intelligence
## 4 Intelligence
## 5 Intelligence
## 6 Intelligence
## 7 Intelligence
## 8 Intelligence
##                                                                                achievement_measu
## 1
## 2
## 3
## 4                                Average report card grades for language arts, math, social studie
## 5                                                                  Average course grades from lan
## 6                                     GPAs were self reported (e.g., I receive mostly As, Bs, Cs,
## 7         Calculations test from the Woodcock Johnson Tests of Achievement-III (Woodcock, McGrew, &
## 8 Passage Comprehension test from the Woodcock Johnson Tests of Achievement-III (Woodcock, McGrew, &
##   academic_achievement_measure_type lab_based published    es_type calculation
## 1                               GPA        no        no continuous Pearson's r
## 2                               GPA        no       yes continuous Pearson's r
## 3                               GPA        no       yes continuous Pearson's r
## 4                               GPA        no       yes continuous Pearson's r
## 5                               GPA        no       yes continuous Pearson's r
## 6                               GPA        no        no continuous Pearson's r
## 7                 Standardized test       yes       yes continuous Pearson's r
## 8                 Standardized test       yes       yes continuous Pearson's r
##       variance adjusted_variance is_significant      r growth_m growth_sd
## 1 0.0052529786      0.0052529786              N -0.060       NA        NA
## 2 0.0006538359      0.0009817362              Y  0.100       NA        NA
## 3 0.0005924735      0.0008896004              Y  0.240       NA        NA
## 4 0.0010198045      0.0010198045              Y  0.210       NA        NA
## 5 0.0021863680      0.0021863680              N  0.090       NA        NA
## 6 0.0116255815      0.0116255815              N -0.010       NA        NA
## 7 0.0012999276      0.0024958790              Y  0.118       NA        NA
## 8 0.0012687748      0.0024345352              Y  0.139       NA        NA
##   other_m other_sd cohen_d rpb rb calculated_r
## 1      NA       NA      NA  NA NA           NA
## 2      NA       NA      NA  NA NA           NA
## 3      NA       NA      NA  NA NA           NA
## 4      NA       NA      NA  NA NA           NA
## 5      NA       NA      NA  NA NA           NA
## 6      NA       NA      NA  NA NA           NA
## 7      NA       NA      NA  NA NA           NA
## 8      NA       NA      NA  NA NA           NA
##
## 1
```

```
## 2
## 3
## 4 the authors of the study used a measure where greater incremental theory was associated with lower
## 5
## 6                          the authors also reported Time 2 data, we used Time 1 because this was pr
## 7
## 8
##        yi     vi
## 1 -0.0601 0.0053
## 2  0.1003 0.0007
## 3  0.2448 0.0007
## 4  0.2132 0.0011
## 5  0.0902 0.0022
## 6 -0.0100 0.0119
## 7  0.1186 0.0013
## 8  0.1399 0.0013
```

```
df3_develop$development_stage <- droplevels(df3_develop$development_stage)
levels(df2_develop$development_stage)
```

```
## [1] "Adolescents" "Adults"      "Children"
```

```
meta1_multi_develop <- rma.mv(yi = yi,
                              V = vi,
                              random = list(~1 | study_id/es_id),
                              data = df3_develop,
                              mods = ~factor(development_stage)-1)
summary(meta1_multi_develop)
```

```
##
## Multivariate Meta-Analysis Model (k = 265; method: REML)
##
##    logLik   Deviance        AIC        BIC       AICc
##   93.3402  -186.6803  -176.6803  -158.8386  -176.4459
##
## Variance Components:
##
##             estim    sqrt  nlvls  fixed          factor
## sigma^2.1  0.0105  0.1026    123     no        study_id
## sigma^2.2  0.0040  0.0634    265     no  study_id/es_id
##
## Test for Residual Heterogeneity:
## QE(df = 262) = 5395.4029, p-val < .0001
##
## Test of Moderators (coefficients 1:3):
## QM(df = 3) = 100.5454, p-val < .0001
##
## Model Results:
##
##                                    estimate      se    zval    pval    ci.lb
## factor(development_stage)Adolescents    0.1465  0.0173  8.4437  <.0001   0.1125
## factor(development_stage)Adults         0.0058  0.0183  0.3157  0.7523  -0.0301
## factor(development_stage)Children       0.1945  0.0346  5.6271  <.0001   0.1268
##                                     ci.ub
## factor(development_stage)Adolescents  0.1805  ***
```

```
## factor(development_stage)Adults       0.0417
## factor(development_stage)Children     0.2623  ***
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## TO BE DONE: HOW TO COMPARE THE THREE CATEGOREIS (ANOVA WITH CORRECTION FOR MULTIPLE COMPARISON)
```

Meta-analysis 2: hakn = TRUE

Change from Meta-analysis 1 only one thing:

- Apply the hakn correction to control for the uncertainty in our estimate of the between-study heterogeneity.

```
meta2 <- metacor(cor = r,
                 n = n,
                 studlab = reference,
                 data= df2,
                 fixed = FALSE,
                 random = TRUE,
                 method.tau = 'REML',
                 hakn = TRUE,
                 title = "Mindset and Academic Achievement")
meta2
```

```
## Review:     Mindset and Academic Achievement
##
## Number of studies combined: k = 273
## Number of observations: o = 419854
##
##                          COR           95%-CI     t  p-value
## Random effects model 0.1067 [0.0858; 0.1274] 10.00 < 0.0001
##
## Quantifying heterogeneity:
##  tau^2 = 0.0187 [0.0164; 0.0284]; tau = 0.1369 [0.1279; 0.1685]
##  I^2 = 97.0% [96.8%; 97.1%]; H = 5.74 [5.56; 5.92]
##
## Test of heterogeneity:
##        Q d.f. p-value
##  8958.24  272       0
##
## Details on meta-analytical method:
## - Inverse variance method
## - Restricted maximum-likelihood estimator for tau^2
## - Q-profile method for confidence interval of tau^2 and tau
## - Hartung-Knapp adjustment for random effects model
## - Fisher's z transformation of correlations
```

Conclusion: The confidence interval is slightly larger than meta1, and the test is based on a t-distribution instead of a z-distribution. The estimate stays the same.

Applying the correction does not change any conclusion, so I will not apply it in subsequent analyses.

Meta-analysis 3: method.tau = 'DL'

```
meta3 <- metacor(cor = r,
                 n = n,
                 studlab = reference,
                 data= df2,
                 fixed = FALSE,
                 random = TRUE,
                 method.tau = 'DL',
                 hakn = FALSE,
                 title = "Mindset and Academic Achievement")
meta3
```

```
## Review:     Mindset and Academic Achievement
##
## Number of studies combined: k = 273
## Number of observations: o = 419854
##
##                          COR            95%-CI    z  p-value
## Random effects model 0.1074 [0.0843; 0.1304] 9.07 < 0.0001
##
## Quantifying heterogeneity:
##  tau^2 = 0.0271; tau = 0.1645; I^2 = 97.0% [96.8%; 97.1%]; H = 5.74 [5.56; 5.92]
##
## Test of heterogeneity:
##        Q d.f. p-value
##  8958.24  272       0
##
## Details on meta-analytical method:
## - Inverse variance method
## - DerSimonian-Laird estimator for tau^2
## - Fisher's z transformation of correlations
```

Meta-analysis 4: method.tau = 'PM'

```r
meta4 <- metacor(cor = r,
                 n = n,
                 studlab = reference,
                 data= df2,
                 fixed = FALSE,
                 random = TRUE,
                 method.tau = 'PM',
                 hakn = FALSE,
                 title = "Mindset and Academic Achievement")
meta4
```

```
## Review:     Mindset and Academic Achievement
##
## Number of studies combined: k = 273
## Number of observations: o = 419854
##
##                         COR           95%-CI    z  p-value
## Random effects model 0.1069 [0.0858; 0.1278] 9.90 < 0.0001
##
## Quantifying heterogeneity:
##  tau^2 = 0.0214 [0.0164; 0.0284]; tau = 0.1463 [0.1279; 0.1685]
##  I^2 = 97.0% [96.8%; 97.1%]; H = 5.74 [5.56; 5.92]
##
## Test of heterogeneity:
##       Q d.f. p-value
##  8958.24  272       0
##
## Details on meta-analytical method:
## - Inverse variance method
## - Paule-Mandel estimator for tau^2
## - Q-profile method for confidence interval of tau^2 and tau
## - Fisher's z transformation of correlations
```

Meta-analysis 5: method.tau = 'EB'

```r
meta5 <- metacor(cor = r,
                 n = n,
                 studlab = reference,
                 data= df2,
                 fixed = FALSE,
                 random = TRUE,
                 method.tau = 'EB',
                 hakn = FALSE,
                 title = "Mindset and Academic Achievement")
meta5
```

```
## Review:     Mindset and Academic Achievement
##
## Number of studies combined: k = 273
## Number of observations: o = 419854
##
##                         COR            95%-CI    z  p-value
## Random effects model 0.1069 [0.0858; 0.1278] 9.91 < 0.0001
##
## Quantifying heterogeneity:
##  tau^2 = 0.0214 [0.0164; 0.0284]; tau = 0.1463 [0.1279; 0.1685]
##  I^2 = 97.0% [96.8%; 97.1%]; H = 5.74 [5.56; 5.92]
##
## Test of heterogeneity:
##       Q d.f. p-value
##  8958.24  272       0
##
## Details on meta-analytical method:
## - Inverse variance method
## - Empirical Bayes estimator for tau^2
## - Q-profile method for confidence interval of tau^2 and tau
## - Fisher's z transformation of correlations
```

Meta-analysis 6: method.tau = 'SJ'

```
meta6 <- metacor(cor = r,
                 n = n,
                 studlab = reference,
                 data= df2,
                 fixed = FALSE,
                 random = TRUE,
                 method.tau = 'SJ',
                 hakn = FALSE,
                 title = "Mindset and Academic Achievement")
meta6
```

```
## Review:     Mindset and Academic Achievement
##
## Number of studies combined: k = 273
## Number of observations: o = 419854
##
##                          COR            95%-CI    z  p-value
## Random effects model 0.1077 [0.0837; 0.1316] 8.75 < 0.0001
##
## Quantifying heterogeneity:
##  tau^2 = 0.0298 [0.0164; 0.0284]; tau = 0.1726 [0.1279; 0.1685]
##  I^2 = 97.0% [96.8%; 97.1%]; H = 5.74 [5.56; 5.92]
##
## Test of heterogeneity:
##       Q d.f. p-value
##  8958.24  272       0
##
## Details on meta-analytical method:
## - Inverse variance method
## - Sidik-Jonkman estimator for tau^2
## - Q-profile method for confidence interval of tau^2 and tau
## - Fisher's z transformation of correlations
```