

Not all cities are the same: variation in animal phenotypes across cities within urban ecology studies

Analyses by Breanna J. Putman

1/11/2022

R Code for the analysis of data for delBarco-Trillo and Putman (2023)

First load and view data

```
#load data
df <- read.csv("ESM_2.csv")

# Create new trait column and make sure it is coded as a factor
df$trait <- df$type.trait.1.behavioral..2.physiological..3.morphological.
df$trait <- as.factor(df$trait)
```

Create a new column for the random effect of "id" (it's required at least for a m-a to calculate the between variance component).

```
# make effect size id column
df$id <- 1:nrow(df)
```

Calculate Effect Sizes, the standardised mean difference ('standardized' effect size, unitless), and add to datasheet

```
library(metafor)
library(dplyr)
SMD <- escalc(measure = "SMD", n1i = df$n.city2, n2i = df$n.city1, m1i =
              df$mean.city2.highest.mean, m2i = df$mean.city1.lowest.mean,
              sd1i = df$sd.city2, sd2i = df$sd.city1)

# Add the effect sizes to the original data set. Save file.
df <- bind_cols(df, SMD)

# make weights, which is 1/vi
df[, "wi"] <- 1/df$vi

names(df) # verify that these worked
```

```
## [1] "paper"
```

```

## [2] "study.id"
## [3] "sp.common.name"
## [4] "sp.scientific.name"
## [5] "phylo"
## [6] "taxon"
## [7] "City1.name"
## [8] "mean.city1.lowest.mean"
## [9] "sd.city1"
## [10] "n.city1"
## [11] "Human.population1"
## [12] "Population.density1"
## [13] "City2.name"
## [14] "mean.city2.highest.mean"
## [15] "sd.city2"
## [16] "n.city2"
## [17] "Human.population2"
## [18] "Population.density2"
## [19] "Absolute.difference.pop.size"
## [20] "Absolute.difference.pop.density"
## [21] "a.priori.difference.between.cities.stated.in.the.paper..e.g..one.larger.or.more.polluted.than."
## [22] "a.priori.diff.binary"
## [23] "dist.between.cities.km"
## [24] "phenotypic.trait"
## [25] "type.trait.1.behavioral..2.physiological..3.morphological."
## [26] "Number.of.cities"
## [27] "n.total"
## [28] "full.reference"
## [29] "citation..original.format."
## [30] "full.reference..original.format."
## [31] "Entry.ID..1st.dataset."
## [32] "Notes"
## [33] "City1.name.1"
## [34] "mean.city1.lowest.mean.1"
## [35] "sd.city1.1"
## [36] "n.city1.1"
## [37] "Human.population"
## [38] "Ref..for.population.estimate"
## [39] "Population.density..individuals.km2."
## [40] "Ref..pop..density"
## [41] "City2.name.1"
## [42] "mean.city2.highest.mean.1"
## [43] "sd.city2.1"
## [44] "n.city2.1"
## [45] "Human.population.1"
## [46] "Source.for.population.estimate"
## [47] "Population.density..individuals.km2..1"
## [48] "Ref..pop..density.1"
## [49] "trait"
## [50] "id"
## [51] "yi"
## [52] "vi"
## [53] "wi"

```

```
write.csv(df, file="Dataset_with_SMD.csv") # save as a new csv file
```

Load Phylogenetic Tree

```
library(ape)
library(phytools)
library(geiger)

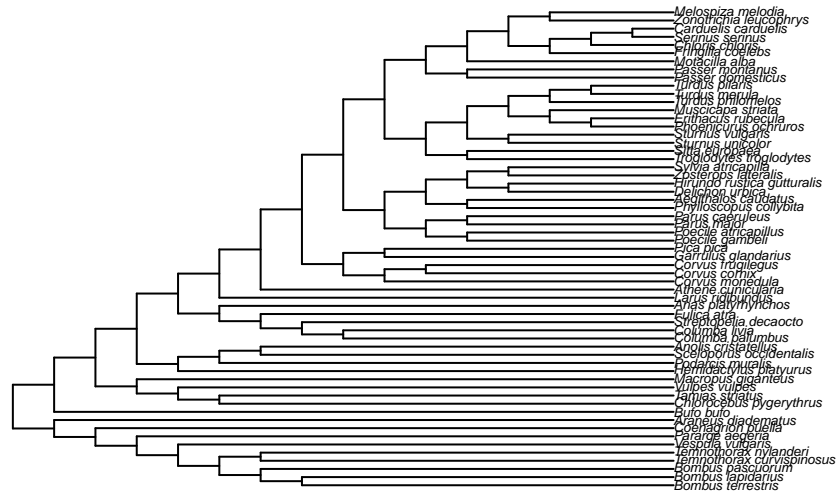
tree <- read.tree(file="Tree_Newick.txt") # load tree
tree<-force.ultrametric(tree, method = "extend")
```

```
## *****
## *                               Note:                               *
## *   force.ultrametric does not include a formal method to      *
## *   ultrametricize a tree & should only be used to coerce     *
## *   a phylogeny that fails is.ultramtric due to rounding --   *
## *   not as a substitute for formal rate-smoothing methods.    *
## *****
```

```
is.ultrametric(tree) # check to see if ultrametric
```

```
## [1] TRUE
```

```
plot(tree, cex = 0.4) # view tree
```



Conduct phylogenetic meta-analytic regression

```
# First, convert the tree to a variance-covariance matrix:
varcor1 = vcv(corBrownian(1, tree), corr=TRUE)

# Run meta-analysis without any moderators
phylo_m <-
  rma.mv(
    yi = yi, # effect size
    V = vi, # variance
    W = wi, # add weights
    random = list(~ 1|phylo, ~ 1|paper, ~ 1|id), # random effects
    R = list(phylo = varcor1),
    method = "REML",
    data = df
  )

# View results
summary(phylo_m)
```

```
##
## Multivariate Meta-Analysis Model (k = 251; method: REML)
##
```

```

##      logLik   Deviance      AIC      BIC      AICc
## -310.6935  621.3869   629.3869   643.4728   629.5502
##
## Variance Components:
##
##      estim      sqrt  nlvls  fixed  factor  R
## sigma^2.1  0.2283  0.4779    59    no   phylo  yes
## sigma^2.2  0.1733  0.4163    53    no   paper  no
## sigma^2.3  0.2455  0.4954   251    no    id    no
##
## Test for Heterogeneity:
## Q(df = 250) = 1645.4443, p-val < .0001
##
## Model Results:
##
## estimate      se      zval      pval      ci.lb      ci.ub
## 0.6526  0.2584  2.5250  0.0116  0.1460  1.1591 *
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```

# Calculating I^2 for Multilevel Models:
# http://www.metafor-project.org/doku.php/tips:i2\_multilevel\_multivariate
W <- diag(1/df$vi)
X <- model.matrix(phylo_m)
P <- W - W %*% X %*% solve(t(X) %*% W %*% X) %*% t(X) %*% W
100 * sum(phylo_m$sigma2) / (sum(phylo_m$sigma2) + (phylo_m$k-phylo_m$p)/sum(diag(P)))

```

```
## [1] 90.21653
```

```

# Calculate how much of total variance is attributed to each heterogeneity
# component separately:
100 * phylo_m$sigma2 / (sum(phylo_m$sigma2) + (phylo_m$k-phylo_m$p)/sum(diag(P)))

```

```
## [1] 31.83304 24.16500 34.21848
```

Perform Model Selection Process to see which moderators of our six are important.

Following instructions from here: http://www.metafor-project.org/doku.php/tips:model_selection_with_glmulti_and_mumin

```

library(glmulti)
library(dplyr)

# First, I renamed variables for clarity/simplicity
df <- rename(df,
             Number_of_Cities = Number.of.cities,
             Distance = dist.between.cities.km,
             Population_Density = Absolute.difference.pop.density,
             Trait = trait,

```

```

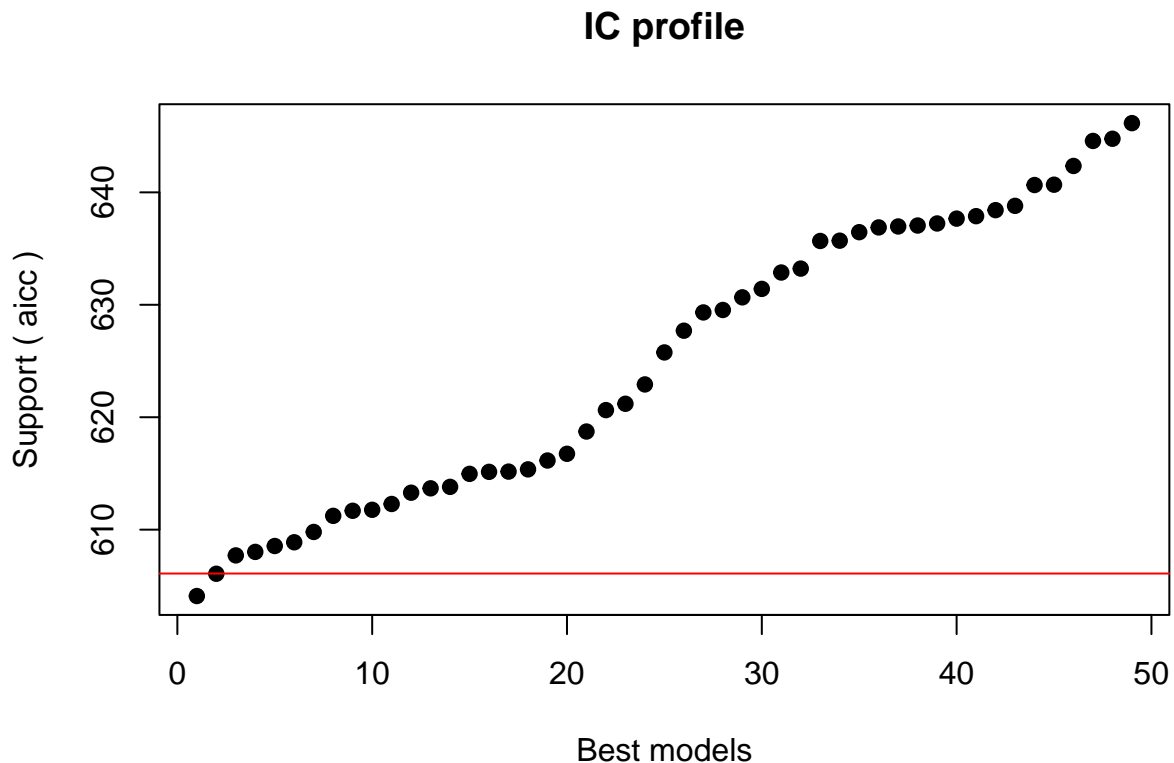
    Apriori_Expectation = a.priori.diff.binary,
    Population_Size = Absolute.difference.pop.size)

# Set up model
rma.glmulti <- function(formula, data, ...)
  rma.mv(formula, vi, wi, random = list( ~ 1|phylo, ~ 1|paper, ~ 1|id),
    data=data, method="ML", ...)

# Run model selection
res <- glmulti(yi ~ factor(Trait) + factor(Apriori_Expectation) + Number_of_Cities
  + Distance + Population_Density + Population_Size,
  data=df, level=1, fitfunction=rma.glmulti, crit="aicc", confsetsize=64)

## Initialization...
## TASK: Exhaustive screening of candidate set.
## Fitting...
##
## After 50 models:
## Best model: yi~1+factor(Trait)+Number_of_Cities+Distance+Population_Density
## Crit= 604.099051710478
## Mean crit= 625.025770810726

```



```
## Completed.
```

```
# print results
print(res)
```

```
## glmulti.analysis
## Method: h / Fitting: rma.glmulti / IC used: aicc
## Level: 1 / Marginality: FALSE
## From 64 models:
## Best IC: 604.099051710478
## Best model:
## [1] "yi ~ 1 + factor(Trait) + Number_of_Cities + Distance + Population_Density"
## Evidence weight: 0.334730790758274
## Worst IC: 646.59948221656
## 3 models within 2 IC units.
## 10 models to reach 95% of evidence weight.
```

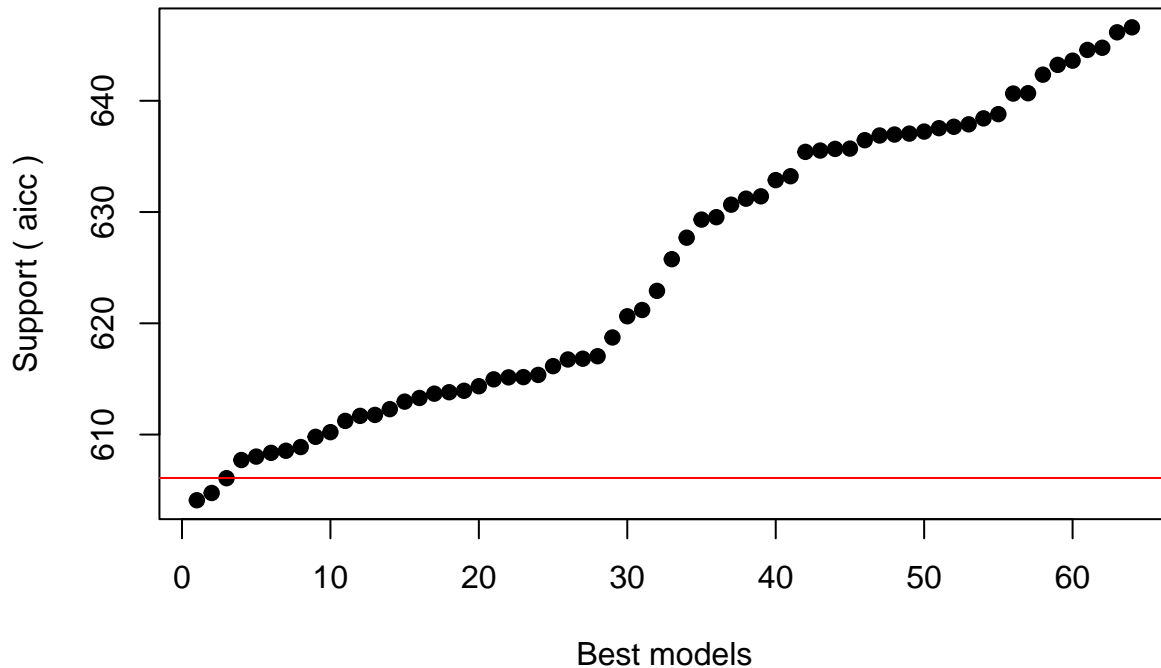
```
# Show top models
top <- weightable(res)
top <- top[top$aicc <= min(top$aicc) + 2,]
print(top) # the Akaike weight for a particular model can be regarded as the probability
```

```
##
## 1 yi ~ 1 + factor(Trait) + Number_of_Cities + Distance + Population_Density
## 2 yi ~ 1 + factor(Trait) + factor(Apriori_Expectation) + Number_of_Cities + Distance + Population_Density
## 3 yi ~ 1 + Number_of_Cities + Distance + Population_Density
## aicc weights
## 1 604.0991 0.3347308
## 2 604.7496 0.2417879
## 3 606.0792 0.1243714
```

```
# that the model is the best model
write.csv(top, "Top_Models.csv") # Save results as .csv file
```

```
# Plot Information Criteria (IC) Profile Plot of all potential models
plot(res)
```

IC profile



Multimodel Inference

```
eval(metafor:::glmulti)
impt_df <- coef(res) # Dataframe with importance values

# Create our own with more informative Column Names and in Order of Importance:
mmi <- as.data.frame(coef(res))
mmi <- data.frame(Estimate=mmi$Est, SE=sqrt(mmi$Uncond), Importance=mmi$Importance,
                 row.names=row.names(mmi))
mmi$z <- mmi$Estimate / mmi$SE
mmi$p <- 2*pnorm(abs(mmi$z), lower.tail=FALSE)
names(mmi) <- c("Estimate", "Std. Error", "Importance", "z value", "Pr(>|z|)")
mmi$ci.lb <- mmi[[1]] - qnorm(.975) * mmi[[2]]
mmi$ci.ub <- mmi[[1]] + qnorm(.975) * mmi[[2]]
mmi <- mmi[order(mmi$Importance, decreasing=TRUE), c(1,2,4:7,3)]
round(mmi, 4)
```

##	Estimate	Std. Error	z value	Pr(> z)	ci.lb
## intrcpt	0.4171	0.2541	1.6417	0.1007	-0.0809
## Number_of_Cities	0.1553	0.0432	3.5943	0.0003	0.0706
## Distance	0.0002	0.0001	1.7923	0.0731	0.0000
## Population_Density	-0.0001	0.0001	-1.1849	0.2361	-0.0002
## factor(Trait)2	-0.1991	0.2211	-0.9007	0.3677	-0.6325


```

## factor(Trait)3          -0.3032    0.2726 -1.1122    0.2661 -0.8375
## factor(Apriori_Expectation)1 -0.0875    0.1762 -0.4965    0.6196 -0.4329
## Population_Size        0.0000    0.0000 -0.3870    0.6988  0.0000
##
##          ci.ub Importance
## intrcpt          0.9150    1.0000
## Number_of_Cities 0.2400    1.0000
## Distance          0.0004    0.9641
## Population_Density 0.0000    0.8473
## factor(Trait)2     0.2342    0.7355
## factor(Trait)3     0.2311    0.7355
## factor(Apriori_Expectation)1 0.2579    0.3795
## Population_Size    0.0000    0.0997

```

```

# Save factor importance values in .csv file:
write.csv(mmi, "factor_importance_renamed_ALL_Jan2022.csv")

# Plot Results
par(mar=c(3,8,3,7))
plot(res, type="s") # relative importance of the various model terms

```

Model-averaged importance of terms



Redo phylogenetic metareg (using REML) with the moderators identified in the top model(s)

Change moderators based on which model you are running.

```

phylo_m2 <-
  rma.mv(
    yi = yi,
    V = vi,
    W = wi,
    random = list(~ 1|phylo, ~ 1|paper, ~ 1|id),
    mods = ~ factor(Trait) + Number_of_Cities + Distance + Population_Density,
    R = list(phylo = varcor1),
    method = "REML",
    data = df)

# View results
summary(phylo_m2)

##
## Multivariate Meta-Analysis Model (k = 251; method: REML)
##
##      logLik   Deviance      AIC      BIC      AICc
## -288.0044   576.0089   594.0089   625.5202   594.7749
##
## Variance Components:
##
##      estim  sqrt  nlvls  fixed  factor  R
## sigma^2.1  0.0487  0.2206    59    no   phylo  yes
## sigma^2.2  0.0624  0.2497    53    no    paper  no
## sigma^2.3  0.2484  0.4984   251    no     id    no
##
## Test for Residual Heterogeneity:
## QE(df = 245) = 1012.3262, p-val < .0001
##
## Test of Moderators (coefficients 2:6):
## QM(df = 5) = 19.3291, p-val = 0.0017
##
## Model Results:
##
##              estimate      se      zval      pval      ci.lb      ci.ub
## intrcpt          0.4522  0.2210   2.0464  0.0407   0.0191  0.8854  *
## factor(Trait)2   -0.2667  0.2190  -1.2176  0.2234  -0.6960  0.1626
## factor(Trait)3   -0.4325  0.2632  -1.6432  0.1003  -0.9485  0.0834
## Number_of_Cities  0.1640  0.0480   3.4205  0.0006   0.0700  0.2580  ***
## Distance          0.0002  0.0001   1.7685  0.0770  -0.0000  0.0003  .
## Population_Density -0.0001  0.0000  -1.3891  0.1648  -0.0002  0.0000
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# Calculating I^2 for Multilevel Models:
# http://www.metafor-project.org/doku.php/tips:i2\_multilevel\_multivariate
W <- diag(1/df$vi)
X <- model.matrix(phylo_m2)
P <- W - W %*% X %*% solve(t(X) %*% W %*% X) %*% t(X) %*% W
100 * sum(phylo_m2$sigma2) / (sum(phylo_m2$sigma2) + (phylo_m2$k-phylo_m2$p)/sum(diag(P)))

```

```
## [1] 82.82203
```

```
# Calculate how much of total variance is attributed to each heterogeneity component separately:  
100 * phylo_m2$sigma2 / (sum(phylo_m2$sigma2) + (phylo_m2$k-phylo_m2$p)/sum(diag(P)))
```

```
## [1] 11.21207 14.37326 57.23670
```

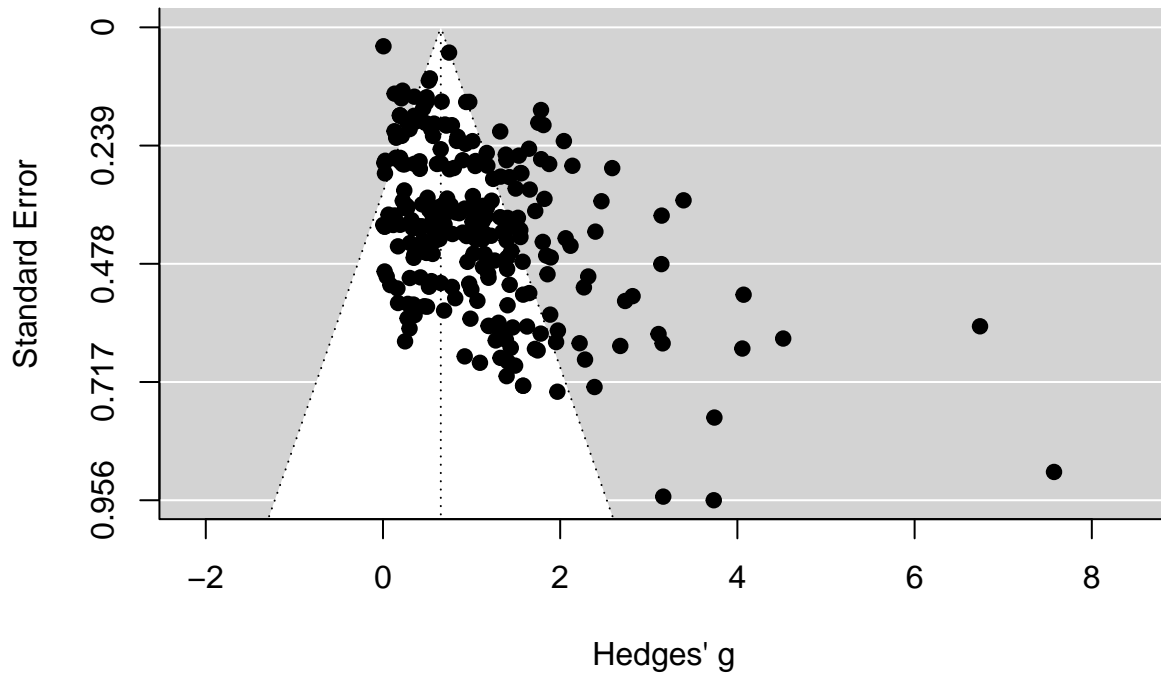
Test for Publication Bias

<https://stat.ethz.ch/pipermail/r-sig-meta-analysis/2018-February/000615.html>

```
# first define model without moderators (cannot handle random effects either)  
metareg <- rma(yi = yi, vi = vi, weights = wi,  
              method = "REML",  
              data = df)  
summary(metareg) # view model results
```

```
##  
## Random-Effects Model (k = 251; tau^2 estimator: REML)  
##  
##      logLik  deviance      AIC      BIC      AICc  
## -357.0731   714.1461   718.1461   725.1891   718.1947  
##  
## tau^2 (estimated amount of total heterogeneity): 0.5512 (SE = 0.0635)  
## tau (square root of estimated tau^2 value):      0.7424  
## I^2 (total heterogeneity / total variability):   88.71%  
## H^2 (total variability / sampling variability):  8.85  
##  
## Test for Heterogeneity:  
## Q(df = 250) = 1645.4443, p-val < .0001  
##  
## Model Results:  
##  
## estimate      se      zval      pval      ci.lb      ci.ub  
##  0.6526  0.1646  3.9636  <.0001  0.3299  0.9753  ***  
##  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# create funnel plot  
funnel(metareg, xlab = "Hedges' g")
```



```
# Egger's test for asymmetry
regtest(metareg, model="rma")
```

```
##
## Regression Test for Funnel Plot Asymmetry
##
## Model:      mixed-effects meta-regression model
## Predictor: standard error
##
## Test for Funnel Plot Asymmetry: z = 2.2992, p = 0.0215
## Limit Estimate (as sei -> 0):  b = 0.1935 (CI: -0.4713, 0.8584)
```

```
# Trim and Fill Method
m_taf <- trimfill(metareg)
m_taf
```

```
##
## Estimated number of missing studies on the left side: 93 (SE = 9.7311)
##
## Random-Effects Model (k = 344; tau^2 estimator: REML)
##
## tau^2 (estimated amount of total heterogeneity): 1.5220 (SE = 0.1311)
## tau (square root of estimated tau^2 value):      1.2337
## I^2 (total heterogeneity / total variability):   94.89%
## H^2 (total variability / sampling variability):  19.57
```

```
##
## Test for Heterogeneity:
## Q(df = 343) = 3308.7616, p-val < .0001
##
## Model Results:
##
## estimate      se      zval      pval      ci.lb      ci.ub
## 0.4278 0.2364 1.8095 0.0704 -0.0356 0.8911 .
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
funnel(m_taf, xlab = "Hedges' g")
```

