Multilevel Meta-Analysis (Pearson r) Tutorial

Hu, M., Koh, P.S., Soh, X.C., Hartanto, A., Majeed, N.M.

Introduction

The following script is designed to perform a multilevel meta-analysis using the metafor, lmerTest and psych packages in R. Specifically, this tutorial utilises Pearson's r and Fisher's Z as the effect size measures, to demonstrate how to convert between the two measures.

The analysis is based on data from Lua et al. (2023) and focuses on the overall relationship between the need for cognition and well-being. The original paper can be found here: https://doi.org/10.1007/s11031-023-10047-w.

The script includes steps for data preparation, effect size calculation, overall effect size computation, forest plot generation, tests for publication bias, and moderation analysis.

The script is structured to be run in RStudio, and it includes comments to guide users through each step of the process.

Setting Up

This section sets up the working environment, installs necessary packages, loads the required libraries, and reads in the data.

If the metafor, lmerTest and psych and packages are not already installed, use the install.packages() function to install it.

Explanation of the Code

- The setwd() function sets the working directory to the location of the script, ensuring that all file paths are relative to the script's location.
- The library() function loads the metafor, lmerTest and psych packages.
- The options() function is used to adjust the display settings, specifically to disable scientific notation and set the number of digits displayed.
- The read.csv() function reads in the data from a CSV file named "NFCWB.csv", which contains the data drawn from Lua et al.(2023).

```
### Set Up ------
# R version 4.5.0

# Set working directory to that of script's current location
setwd(dirname(rstudioapi::getActiveDocumentContext()$path))

# Load packages
library(metafor) # version 4.8-0
```

Loading required package: Matrix

```
## Loading required package: metadat
## Loading required package: numDeriv
##
## Loading the 'metafor' package (version 4.8-0). For an
## introduction to the package please type: help(metafor)
library(lmerTest) # version 3.1-3
## Loading required package: lme4
##
## Attaching package: 'lmerTest'
## The following object is masked from 'package:lme4':
##
##
       lmer
## The following object is masked from 'package:stats':
##
##
       step
library(psych)
                  # version 2.5.3
# Display settings (to disable scientific notation)
options(scipen = 9999, digits = 4)
# Read in data drawn from Lua et al. (2023)
mlmmeta_raw = read.csv("NFCWB.csv")
```

Prepare Data

This section prepares the data for analysis by computing effect sizes for each study and organizing the data frame.

For more information on the escalc() function, refer to ?escalc in R.

- The corr_nfcwb variable has to be reversed for studies with negative well-being, ensuring that the correlation reflects that the lower the correlation, the lower the well-being. The ifelse() function is used to reverse the correlation for studies with negative well-being.
- The measure argument specifies the type of effect size to be calculated, in this case, "ZCOR" (Fisher's Z). ZCOR is used to calculate the effect size from the raw correlation coefficients instead of Pearson's r as Fisher's Z normalises the distribution of the effect sizes. Thus, Fisher's Z would be less affected by the sampling distribution skew.
- The ri variable corr_nfcwb is the column containing the raw correlation coefficients.
- The ni variable sample_size is the column containing the sample sizes for each study.
- Afterwards, the escalc() function computes the effect sizes (yi) and their corresponding sampling variances (vi) for each study.
- The publication type is categorised into "Published" and "Unpublished" based on the type of publication (e.g., journal article, conference paper, thesis/dissertation).
- The mlmmeta data frame is reordered to prepare it for visualization. The order() function sorts the rows first by publication_type (grouping the same publication types together) and then by yi (arranging

effect sizes from smallest to largest within each publication type). Do note that the comma before] is required as it tells R to keep all columns while reordering only the rows according to the specified order.

```
### Prepare Data -----
# Clean data file (reverse correlation for negative well-being)
mlmmeta_raw$corr_nfcwb = with(mlmmeta_raw,
ifelse(wellbeing_category == "Negative well-being", -corr_nfcwb, corr_nfcwb))
# Compute effect sizes for each study
mlmmeta = escalc(
  # Type of effect size measure
  measure = "ZCOR",
  # Column for raw correlation coefficients
  ri = corr_nfcwb,
  # Column for sample sizes
  ni = sample size,
  # Specify data.frame that the information will be extracted from
  data = mlmmeta raw
# Categorise publication type into "published" and "unpublished"
# Published: Journal articles
# Unpublished: Conference, Panel Data, Thesis/dissertation, Unpublished data
mlmmeta$publication_type = ifelse(
  mlmmeta$publication_type == "Journal article",
  "Published",
  "Unpublished")
# Order the data frame based on publication type and effect sizes (yi)
mlmmeta = mlmmeta[order(mlmmeta$publication_type, mlmmeta$yi), ]
```

Computing the Overall Effect Size

This section estimates the overall effect size using the rma.mv() function from the metafor package.

Explanation of the Multilevel Meta-Analysis Code

- The rma.mv() function is used to compute the overall effect size, accounting for the nested structure of the data.
- The random argument specifies the random effects structure, where ~ 1 | sample_id/meta_id, indicates that random effects are nested within the sample and meta ID.
- The yi and vi arguments specify the effect size estimates and their corresponding sampling variances, respectively.
- The data argument specifies the data frame containing the effect size estimates and variances.
- The summary() function is used to display the results of the meta-analysis, including the overall effect size estimate and its confidence interval.
- The fisherz2r() function is used to convert the effect size estimates from Fisher's Z to Pearson's r.

```
### Compute Overall Effect Size --
# Effect size estimates
mlmmetaresults = rma.mv(
  # Effect size estimates
 yi = yi,
  # Sampling variances
 V = vi,
  # Include random effects for grouping variable (i.e., sample)
  random = ~ 1 | sample_id/meta_id,
  # Specify where to get the data from
  data = mlmmeta
)
# summary function used to provide detailed results of the meta-analysis
summary(mlmmetaresults)
##
## Multivariate Meta-Analysis Model (k = 108; method: REML)
##
##
                            AIC
                                      BIC
                                               AICc
    logLik Deviance
##
   40.9843 -81.9687 -75.9687 -67.9502 -75.7357
##
## Variance Components:
##
##
                             nlvls fixed
                                                       factor
               estim
                        sqrt
                                52
## sigma^2.1 0.0076 0.0873
                                                    sample_id
                                       no
## sigma^2.2
             0.0156 0.1250
                                108
                                       no sample_id/meta_id
##
## Test for Heterogeneity:
## Q(df = 107) = 1573.7169, p-val < .0001
## Model Results:
## estimate
                 se
                       zval
                               pval
                                      ci.lb
                                              ci.ub
##
    0.1977 0.0200 9.8908 <.0001 0.1585 0.2369
##
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.05 '.' 0.1 ' ' 1
# Convert from Fisher's Z to Pearson's r
mlmmetaresults$b |> fisherz2r()
             [,1]
## intrcpt 0.1952
```

Forest Plot

This section generates a forest plot to visually represent the effect sizes and confidence intervals for each study included in the meta-analysis.

The forest plot is created using the forest() function from the metafor package.

The plot includes the following features:

• Arrangement of studies by effect sizes

- Sample size information for need for cognition and well-being group
- Custom headers for the plot
- Custom labels for the studies

Saving the forest plot as a PDF file allows for easy sharing and presentation, and allows adjustment to the plot's dimensions.

For reference, the forest plot will appear as shown on page 7 of this handbook.

- The cairo_pdf() function starts the graphics device driver to create PDF files, and the file argument specifies the name of the file. Specifically, the cairo_pdf function is used for font compatibility and in this case, it is for the author's names.
- The width and height arguments adjust the dimensions of the PDF file.
- The forest() function is used to create the forest plot, and the mlmmetaresults object contains the results of the meta-analysis.
- The order argument specifies the arrangement of studies, with "obs" indicating that the studies should be arranged by effect sizes. To organise by column, replace "obs" with the specific column in the data frame.
- The ylim argument sets the y-axis limits for the plot.
- The ilab argument is used to add extra columns of information to the forest plot beyond just the effect sizes. Here, we are adding sample size data for cognition and well-being group into the forest plot.
- The ilab.xpos argument specifies where the sample size columns appear horizontally on the plot. The negative values position the columns on the left side of the plot, just after the study labels.
- The slab argument is used to label each effect size with its respective study.
- The paste() function creates the label by combining the author and year columns. The sep argument specifies the separator between author and the year label, which is set to "," in this case. Hence, the label will be in the format "Author(s), Year" (e.g., "Smith et al., Study 1").
- The xlim argument sets the x-axis limits for the plot.
- The alim argument sets the confidence interval limits, and the steps argument determines the number of intervals in the x-axis.
- The efac argument controls the size of the diamond shapes that represent the effect sizes in the plot.
- The header argument is set to FALSE to allow for manual specification of headers.
- The xlab argument specifies the confidence interval label for the forest plot, in this case, "Fisher's Z"
- The text() function is used to manually include text within the plot, such as the "Author(s) Year" header and specific sample size column headers.
- The x and y arguments in the text() function adjust the position of the headers, with the x argument specifying the horizontal arrangement of the columns and the y argument specifying the vertical arrangement of the columns.
- The font argument adjusts the font size.
- The dev.off() function is used to close the graphics device and finalise the plot as a saved file.

```
### Forest Plot -----
# Save the forest plot as a PDF file
```

```
# Name the pdf file of the forest plot
cairo_pdf(file = "NFCWBforestplot.pdf", width = 14, height = 35)
# Start creating the forest plot itself
# Specify dataset
forest(
  mlmmetaresults,
  # Arrangement of studies
  order = "obs",
  # Add y-axis limits
  ylim = c(-3, 111),
  # Add sample size information for need for cognition and well-being group
  ilab = sample_size,
  ilab.xpos = -3,
  # Label studies on the forest plot
  slab = paste(author, year, sep = ", "),
  # Add x-axis limits
  xlim = c(-5, 3),
  # Add confidence interval limits
  # Adjust intervals based on the number of steps
  alim = c(-1.5, 1.5),
  steps = 7,
  # Change size of effect size polygons
  efac = 0.3,
  # Show (TRUE) or hide (FALSE) default headers
  # Hide when we want to manually specify our own headers
 header = FALSE,
  # Add label for confidence interval, in this case, "Fisher's Z"
 xlab = "Fisher's Z"
# For the following lines of code,
# Use text function to manually include text within the plot
# Add "Author(s) Year" header
text(x = -4.6, y = 110, "Author(s) Year", font = 2)
# Add "Sample Size" header
text(x = -3, y = 110, "Sample Size", font = 2)
# Add "r [95% CI]" header
text(x = 2.7, y = 110, "Z [95\% CI]", font = 2)
# Close the forest plot and finalise it as a saved file
```

dev.off()

pdf ## 2

author(s) Year	Sample Size		Z [95% (
Roiser et al., 2009 Osberg et al., 2008	30 98	⊢	-0.44 [-0.81, -0.0 -0.13 [-0.33, 0.0
Jiberg et al., 2008 Hüttel et al., 2020	98 1075		-0.13 [-0.33, 0.0 -0.10 [-0.16, -0.0
Hartanto, 2021b	253	⊢ ■-1	-0.07 [-0.19, 0.0
Maloney & Retanal, 2019	210	H 	-0.05 [-0.19, 0.0
Allan & Shearer, 2012 Napier & Jost, 2008	316 1142	 ii iii	-0.02 [-0.13, 0.0 0.00 [-0.06, 0.0
Hartanto, 2021a	261	'₹' ‡ 	0.00 [-0.12, 0.1
Hartanto, 2021b	253	⊢ •	0.02 [-0.10, 0.1
Allan & Shearer, 2012 Henning & Vorderer, 2001	316	 = + = 	0.04 [-0.07, 0.1 0.05 [-0.06, 0.1
Allan & Shearer, 2012	320 316		0.05 [-0.06, 0.1
Gentile et al., 2019	496	⊬æ⊣	0.05 [-0.04, 0.1
Bye, 2012	326	 ■ 	0.05 [-0.06, 0.1
Karnaze & Levine, 2020 Bye, 2012	282 275	H = 1 H = 1	0.06 [-0.06, 0.1 0.06 [-0.06, 0.1
owery, 2007	60	⊢	0.07 [-0.19, 0.3
Hartanto, 2021b	253	!= -1	0.07 [-0.05, 0.1
rong, 2021 Strobel et al., 2017	185	 • 	0.07 [-0.08, 0.2 0.08 [-0.06, 0.2
Shayyer et al., 2019	202 182	 	0.08 [-0.06, 0.2
Vezlek et al., 2001	112	↓	0.09 [-0.10, 0.2
Shchebetenko et al., 2019	1008	l a l	0.09 [0.03, 0.1
Miceli, 2012 Hartanto, 2021b	163	<u>†</u>	0.09 [-0.06, 0.2 0.09 [-0.03, 0.2
ISS Dataset, 2008	253 6802	 ■ ■	0.09 [0.07, 0.1
Prohaska et al., 2017	192		0.10 [-0.04, 0.2
Owyer, 2008	207	1 ■ 1	0.10 [-0.04, 0.2
lye, 2012 lye, 2012	275	-	0.10 [-0.02, 0.2
Sye, 2012 Cavazos & Campbell, 2008	275 703	 	0.11 [=0.01, 0.2 0.12 [0.05, 0.1
Vezlek et al., 2001	112	T=T	0.12 [-0.05, 0.1
Grass & Strobel, 2019	188	' ■ 1'	0.12 [-0.02, 0.2
Osberg et al., 2008	99	 ■ 	0.13 [-0.07, 0.3
lye, 2012 erry, 2001	326	- - - - -	0.13 [0.02, 0.2 0.13 [0.01, 0.2
erry, 2001 Sauthier et al., 2006	263 192		0.13 [0.01, 0.2
azdani & Siedlecki, 2021	4354	i	0.14 [0.11, 0.1
be, 2018	601	₽₽H	0.14 [0.06, 0.2
e, 2010	436	 ■	0.14 [0.05, 0.2
lartanto, 2021a lartanto, 2021a	261 261	}- = -1	0.14 [0.02, 0.2 0.14 [0.02, 0.2
lartanto, 2021a lartanto, 2021a	261 261	! -	0.14 [0.02, 0.2
arnaze & Levine, 2020	282	⊢= -i	0.15 [0.03, 0.2
be, 2018	599	H all	0.15 [0.07, 0.2
ye, 2012	275	⊢=- !	0.15 [0.03, 0.2
erry, 2001 erry, 2001	263 263	⊢= -I	0.15 [0.03, 0.2 0.15 [0.03, 0.2
rass & Strobel, 2019	188		0.15 [0.01, 0.3
lüttel et al., 2020	1070	I ≡ I	0.16 [0.10, 0.2
yree, 2003	96	⊬ ■─	0.17 [-0.03, 0.3
e, 2010	436	 	0.18 [0.09, 0.2
lartanto, 2021b iye & Pushkar, 2009	253 351	- - - - - 	0.18 [0.06, 0.3 0.19 [0.09, 0.3
Serstorf et al., 2009	784	i -M	0.20 [0.13, 0.2
fartanto, 2021b	253	⊢ ■→	0.20 [0.08, 0.3
Reeves et al., 1995	306	H=1	0.21 [0.10, 0.3
Nezlek et al., 2001 Subramaniam, 2016	112 973	 	0.21 [0.03, 0.4 0.21 [0.15, 0.2
Graham, 2010	154	·= ·	0.22 [0.06, 0.3
Hartanto, 2021b	253	⊢ ∎ ⊣	0.22 [0.10, 0.3
Heintzelman & King, 2016	1026	 	0.23 [0.17, 0.3
Sabbiadini & Greitemeyer, 2016 Khayyer et al., 2019	392 182	 	0.23 [0.13, 0.3 0.23 [0.09, 0.3
mamo.lu, 2003	274	⊢ ■-i	0.23 [0.12, 0.3
akeuchi et al., 2019	1116	l = I	0.23 [0.18, 0.2
Bertrams & Dickhäuser, 2012	150	⊢• →	0.24 [0.08, 0.4
Bleidorn et al., 2020 fartanto, 2021a	415 261	 ■ - ■ 	0.24 [0.15, 0.3 0.24 [0.12, 0.3
fartanto, 2021a	261		0.24 [0.12, 0.3
fartanto, 2021b	253	⊢ ≡ ⊣	0.24 [0.12, 0.3
agby et al., 1986	81	⊢-	0.26 [0.03, 0.4
be, 2018 yree, 2003	602 96	H	0.26 [0.18, 0.3 0.26 [0.05, 0.4
artanto, 2021a	261	—————————————————————————————————————	0.26 [0.13, 0.3
obin & Guadagno, 2019	310	i - i	0.26 [0.14, 0.3
be, 2018	598	H EH	0.28 [0.20, 0.3
owman, 2012 be, 2018	8615 597	■ I = 1	0.29 [0.27, 0.3 0.29 [0.21, 0.3
pe, 2018 artanto, 2021b	597 253		0.29 [0.21, 0.3
owman, 2012	8615	· <u>-</u> ·	0.34 [0.32, 0.3
ree, 2003	96	⊢ •	0.34 [0.14, 0.5
artanto, 2021a trobel et al., 2017	261	⊢	0.34 [0.22, 0.4
trobel et al., 2017 uri & Monroe, 2001	202 85	⊢= -1	0.35 [0.22, 0.4 0.35 [0.14, 0.5
eden et al., 2019	210	⊢= -1	0.35 [0.22, 0.4
ree, 2003	96	⊢ ■	0.35 [0.15, 0.5
irass & Strobel, 2019	188		0.35 [0.21, 0.5
artanto, 2021b artanto, 2021b	253 253	├ ड -1	0.35 [0.23, 0.4 0.37 [0.24, 0.4
artanto, 2021b artanto, 2021a	253 261		0.37 [0.24, 0.4
artanto, 2021b	253	⊢= -i	0.38 [0.25, 0.5
owman, 2012	8615	. •	0.39 [0.37, 0.4
ezlek et al., 2001 artanto, 2021b	112 253	├─ड ─┤ ├ ड ─┤	0.40 [0.21, 0.5 0.41 [0.29, 0.5
owman, 2012	253 8615	•	0.41 [0.29, 0.5
artanto, 2021a	261	H a H	0.44 [0.31, 0.5
artanto, 2021a	261	⊢ ≡ ⊣	0.45 [0.33, 0.5
artin et al., 2020	212	H	0.46 [0.32, 0.6
be, 2018 artanto, 2021b	601 253	} = -1	0.51 [0.43, 0.5 0.51 [0.39, 0.6
ree, 2003	253 96		0.51 [0.39, 0.6
enny & Banks, 2015	60	, <u>, , , , , , , , , , , , , , , , , , </u>	0.55 [0.29, 0.8
artanto, 2021a	261	⊢ ≡ ⊣	0.55 [0.43, 0.6
alama-Younes, 2016	194	⊢= -1	0.58 [0.43, 0.7
raham, 2010 outinho & Woolery, 2004	154 157	⊢= -1	0.58 [0.42, 0.7 0.73 [0.57, 0.8
ree, 2003	157 96		0.73 [0.57, 0.8
		' - '	
andom-Effects Model		•	0.20 [0.16, 0.2
		· · · · · · · · · · · · · · · · · · ·	

```
## null device
## 1
```

Tests for Publication Bias

This section performs tests for publication bias, including a funnel plot and Egger's test.

For multilevel meta-analysis, we do not recommend conducting a rank correlation test as it is prone to Type 1 error.

Researchers may refer to this article by Fernández-Castilla et al. (2019) on a detailed discussion of the limitations of rank correlation tests in multilevel meta-analysis, as well as an overview on other publication bias tests: https://doi.org/10.1080/00220973.2019.1582470:

Funnel Plot

The funnel plot visually represents the distribution of effect sizes and their standard errors, allowing for the identification of potential publication bias.

Saving the funnel plot as a PDF file allows for easy sharing and presentation, and allows adjustment to the plot's dimensions

For reference, the funnel plot will appear as shown on page 10 of this handbook.

- The pdf() function starts the graphics device driver to create PDF files, and the file function specifies the name of the file.
- The width and height arguments adjust the dimensions of the PDF file.
- The par() function is used to adjust the margins of the funnel plot, with the mar argument specifying the bottom, left, top, and right margins.
- The funnel() function is used to create the funnel plot, and the mlmmetaresults object contains the results of the meta-analysis.
- The legend argument specifies whether to include a legend in the plot. TRUE indicates that a legend should be included, FALSE indicates that it should not.
- The xlab argument specifies the confidence interval label for the funnel plot, in this case, "Fisher's Z".
- The dev.off() function is used to close the graphics device and finalize the plot as a saved file.

```
### Tests for Publication Bias ------
# Funnel Plot #

# Save the funnel plot as a PDF file
# Name the pdf file of the funnel plot
# Adjust the width and height of the pdf file
pdf(file = "NFCWBfunnelplot.pdf", width = 8, height = 5)

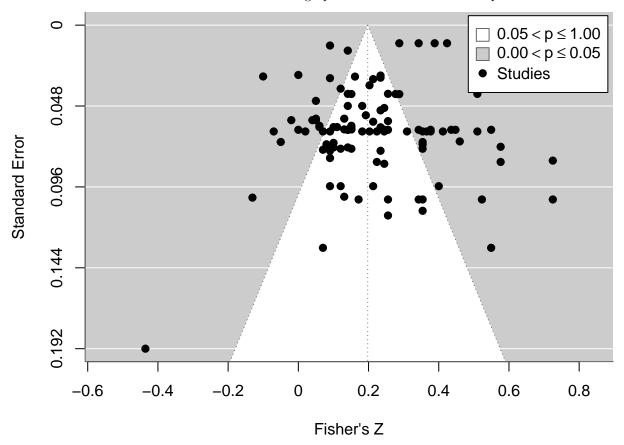
# Adjust margins of the funnel plot
# Set the bottom, left, top, and right margins
par(mar = c(4, 4, 0.3, 1))
# Create the funnel plot
funnel(mlmmetaresults, legend = TRUE, xlab = "Fisher's Z")
```

pdf ## 2

Optional: Saving the Funnel Plot as a Separate File

Saving the funnel plot as a PDF file allows for easy sharing and presentation, and allows adjustment to the plot's dimensions

- To save the funnel plot as a PDF file, the plotting code can be enclosed within pdf() and dev.off() functions:
- The pdf() function starts the graphics device driver to create PDF files, and the file function specifies the name of the file.
- The width and height arguments adjust the dimensions of the PDF file.
- Following the pdf() function, the same code used to create the funnel plot is repeated to generate the plot within the PDF file. The graphical output will be directed to the specified PDF file instead of the RStudio plotting window.
- The dev.off() function is used to close the graphics device and finalize the plot as a saved file.



Egger's Test

The Egger's test is a statistical test that quantifies the degree of asymmetry in the funnel plot, providing a more formal assessment of publication bias.

- The lmer() function is used to fit a linear mixed-effects model, where the effect size weighted by the standard error is predicted by the intercept and the inverse of the corrected standard error. metafor::rma.mv does not have a weights argument, and metafor::regtest does not support rma.mv objects. For three (or more) level meta-analysis, use lmerTest::lmer instead.
- The I(yi / vi) expression indicates that the effect size (yi) is divided by the standard error (vi), which is used to weight the effect sizes in the model.
- The I(1 / vi) expression indicates that the inverse of the standard error is included in the model as a predictor.
- The 1 | sample_id expression indicates that random intercepts are included for each lab, accounting for the nested structure of the data.
- The data argument specifies the dataset to be used for the analysis.
- The summary() function provides the results of the Egger's test, including the slope estimate and its significance.

```
### Tests for Publication Bias -----
# Eggers' Test #
Imer(
    # g weighted by SE is predicted by intercept and inverse SE
    # with random intercept by sample
    I(yi / vi) ~ 1 + I(1 / vi) + (1 | sample_id),
    data = mlmmeta
) |>
    # Estimate of interest is the intercept
    summary(correlation = TRUE)

## Warning: Some predictor variables are on very different scales: consider
## rescaling
## Warning: Some predictor variables are on very different scales: consider
## rescaling
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: I(yi/vi) ~ 1 + I(1/vi) + (1 | sample_id)
      Data: mlmmeta
##
##
## REML criterion at convergence: 1427
##
## Scaled residuals:
##
              10 Median
                            3Q
      Min
                                  Max
## -4.537 -0.103 0.022 0.110 5.077
##
## Random effects:
## Groups
              Name
                          Variance Std.Dev.
## sample_id (Intercept) 42221
                                   205
## Residual
                          14832
                                   122
```

```
## Number of obs: 108, groups: sample_id, 52
##
## Fixed effects:
##
             Estimate Std. Error
                                                          Pr(>|t|)
                                      df t value
                                          -1.16
## (Intercept) -41.0231
                        35.3050 50.5712
                                                              0.25
                          0.0202 45.8603
                                          ## I(1/vi)
               0.2564
## Signif. codes: 0 '***' 0.001 '**' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
          (Intr)
## I(1/vi) -0.420
## fit warnings:
## Some predictor variables are on very different scales: consider rescaling
```

Moderation Analysis

This section performs moderation analysis to explore the influence of categorical moderators on the effect sizes.

Explanation of the Code

- The rma.mv() function is used to compute the multilevel meta-analysis with moderators.
- The yi and vi arguments specify the effect size estimates and their corresponding sampling variances, respectively.
- The random argument specifies the random effects structure, where ~ 1 | sample_id/meta_id indicates that random intercepts are included for both the lab and individual studies, accounting for the nested structure of the data.
- The subset argument is used to specify the subset of data for each moderator analysis.
- The data argument specifies the data frame containing the effect size estimates and variances.

Note that if convergence issues arise, the control function can be used to address it. Researchers may also refer more to the metafor package documentation for more information on how to address convergence issues.

```
### Moderation Analysis -----
# Categorical Variable (i.e., publication type)
rma.mv(
  yi = yi,
  V = vi,
  random = ~ 1 | sample_id/meta_id,
  # Specify categorical moderator (i.e., Published articles)
  subset = (publication_type == "Published"),
  data = mlmmeta,
  # To address convergence issues (if it exists)
  control = list(rel.tol=1e-8)
)
##
## Multivariate Meta-Analysis Model (k = 54; method: REML)
## Variance Components:
##
```

```
factor
              estim
                        sqrt nlvls fixed
                                 37
## sigma^2.1 0.0158 0.1257
                                       nο
                                                    sample_id
## sigma^2.2 0.0103 0.1016
                                 54
                                       no
                                           sample_id/meta_id
##
## Test for Heterogeneity:
## Q(df = 53) = 1058.6768, p-val < .0001
## Model Results:
##
## estimate
                       zval
                               pval
                                      ci.lb
                                              ci.ub
    0.1915 0.0275
                    6.9686
                            <.0001
                                    0.1376
                                            0.2453
##
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.05 '.' 0.1 ' ' 1
rma.mv(
  yi = yi,
  V = vi,
 random = ~ 1 | sample_id/meta_id,
  # Specify categorical moderator (i.e., Unpublished articles)
  subset = (publication_type == "Unpublished"),
  data = mlmmeta,
  # To address convergence issues (if it exists)
  control = list(rel.tol=1e-8)
)
##
## Multivariate Meta-Analysis Model (k = 54; method: REML)
## Variance Components:
##
                             nlvls fixed
##
                                                       factor
               estim
                        sqrt
                     0.0633
## sigma^2.1
             0.0040
                                 15
                                       no
                                                    sample_id
             0.0170 0.1304
                                 54
## sigma^2.2
                                       no
                                           sample_id/meta_id
## Test for Heterogeneity:
## Q(df = 53) = 351.3640, p-val < .0001
##
## Model Results:
##
## estimate
                                      ci.lb
                 se
                       zval
                               pval
                                              ci.ub
##
     0.2126
           0.0289
                    7.3678
                            <.0001
                                    0.1561
                                            0.2692
##
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.05 '.' 0.1 ' ' 1
```

Forest Plot of Moderators

This section generates a forest plot that includes moderators to visually represent the effect sizes and confidence intervals for each study included in the meta-analysis.

The forest plot includes the following features:

- Arrangement of studies by effect sizes and types of moderators
- Sample size information for need for cognition and well-being group

- Custom headers for the plot
- Custom labels for the studies
- Summary effect sizes for each moderator

Saving the forest plot as a PDF file allows for easy sharing and presentation, and allows adjustment to the plot's dimensions.

For reference, the forest plot of moderators will appear as shown on page 17 of this handbook.

- The cairo_pdf() function starts the graphics device driver to create PDF files, and the file argument specifies the name of the file. Specifically, the cairo_pdf function is used for font compatibility and in this case, it is for the author's names.
- The width and height arguments adjust the dimensions of the PDF file.
- The forest() function is used to create the forest plot, and the mlmmetaresults object contains the results of the meta-analysis.
- The rows argument specifies the arrangement of studies by publication type, and in ascending order of effect sizes per publication type.
- The ylim argument sets the y-axis limits for the plot.
- The ilab argument is used to add extra columns of information to the forest plot beyond just the effect sizes. Here, we are adding sample size data for cognition and well-being group into the forest plot.
- The ilab.xpos argument specifies where the sample size columns appear horizontally on the plot. The negative values position the columns on the left side of the plot, just after the study labels.
- The slab argument is used to label each effect size with its respective study.
- The paste() function creates the label by combining the author and year columns. The sep argument specifies the separator between author and the year label, which is set to "," in this case. Hence, the label will be in the format "Author(s), Year" (e.g., "Smith et al., Study 1").
- The xlim argument sets the x-axis limits for the plot.
- The alim argument sets the confidence interval limits, and the steps argument determines the number of intervals in the x-axis.
- The efac argument controls the size of the diamond shapes that represent the effect sizes in the plot.
- The header argument is set to FALSE to allow for manual specification of headers.
- The xlab argument specifies the label for the confidence interval, in this case, "Hedge's g".
- The text() function is used to manually include text within the plot, such as the "Author(s) Year" header and specific sample size column headers.
- The x and y arguments in the text() function adjust the position of the headers, with the x argument specifying the horizontal arrangement of the columns and the y argument specifying the vertical arrangement of the columns.
- The font argument adjusts the font size.
- The post argument specifies the position of the text relative to the specified coordinates
- The rma.mv() function is used to perform moderation analysis for each publication type, with the subset argument specifying the subset of data for each category.
- The rest.j, res.t, and res.c variables store the results of the moderation analysis for journal articles, thesis/dissertations, and conference papers, respectively.

- The subset argument is used to specify the subset of data for categorical moderators, allowing for separate analyses for each category.
- The addpoly() function is used to add summary effect sizes for each of the moderators, with the row argument specifying the position of the summary in the plot.
- The dev.off() function is used to close the graphics device and finalise the plot as a saved file.

```
### Forest Plot of Moderators -----
# Save the forest plot as a PDF file
# Name the pdf file of the forest plot
# cairo_pdf function used for font compatibility
# Adjust the width and height of the pdf file
cairo_pdf(file = "NFCWBforestplotwithmod.pdf", width = 13, height = 35)
# Start creating the forest plot itself
# Specify dataset
forest(
 mlmmetaresults,
  # Manually arrange effect sizes by publication type
  # - Unpublished: Rows 108 to 51
  # - Published: Rows 50 to 48
  # The arrangement must consider spacing and must end at row 2
  rows = c(112:40, 36:2),
  # Add y-axis limits
 ylim = c(-3, 116),
  # Add sample size information for need for cognition and well-being group
  # Values indicate the x-axis position of the sample size columns
  ilab = sample_size,
  ilab.xpos = -4.2,
  # Label studies on the forest plot
  slab = paste(author, year, sep = ", "),
  # Add x-axis limits
  xlim = c(-7, 4),
  # Add confidence interval limits
  # Adjust intervals based on the number of steps
  alim = c(-1.5, 1.5),
  steps = 7,
  # Change size of effect size polygons
  efac = 0.3.
  # Remove headers (if any), for manual input
  header = FALSE,
  # Add label for confidence interval, in this case, "Fisher's Z"
  xlab = "Fisher's Z"
)
```

```
# For the following lines of code,
# Use text function to manually include text within the plot
# Add text labels for moderator (type of publication)
# Labels for different publication types (Moderator Analysis)
\# - "Unpublished" (Unpublished data, Panel Data, Thesis/Dissertations) at y = 37
\# - "Published" (Journal Articles, Conference) at y = 113
text(
 x = -7,
 y = c(37, 113),
 pos = 4,
 c("Unpublished", "Published"),
 font = 2
# Moderation analysis
res.p = rma.mv(
 уi,
  νi,
 random = ~ 1 | sample_id/meta_id,
 subset = (publication_type == "Published"),
 data = mlmmeta,
  # To address convergence issues (if it exists)
  control = list(rel.tol=1e-8)
)
res.u = rma.mv(
  yi,
  vi,
  random = ~ 1 | sample_id/meta_id,
 subset = (publication_type == "Unpublished"),
 data = mlmmeta,
 # To address convergence issues (if it exists)
  control = list(rel.tol=1e-8)
# Add summary effect sizes for each of the moderators
addpoly(res.u, row = 1) # summary effect for "Unpublished" group
addpoly(res.p, row = 39) # summary effect for "Published" group
# Add"Author(s) Year" header
text(x = -6.4, y = 115, "Author(s) Year", font = 2)
# Add "Sample Size" header
text(x = -4.0, y = 115, "Sample Size", font = 2)
# Add "r [95% CI]" header
text(x = 3.6, y = 115, "Z [95\% CI]", font = 2)
# Close the forest plot and finalise it as a saved file
dev.off()
## pdf
```

2

Sample Size		Z [95% CI]
30	⊢ •−−1	-0.44 [-0.81, -0.06]
	H=+	-0.13 [-0.33, 0.07]
	⊫ : + = -1	-0.10 [-0.16, -0.04] -0.05 [-0.19, 0.09]
	-	-0.02 [-0.13, 0.09]
1142	•	0.00 [-0.06, 0.06]
	<u>#</u>	0.04 [-0.07, 0.15]
		0.05 [-0.06, 0.16] 0.05 [-0.06, 0.16]
	=	0.05 [-0.04, 0.14]
282	H im l	0.06 [-0.06, 0.18]
202	I ¦■ I	0.08 [-0.06, 0.22]
		0.09 [-0.06, 0.24] 0.09 [-0.10, 0.28]
		0.09 [0.03, 0.15]
192	Ĥ	0.10 [-0.04, 0.24]
207	∳= -1	0.10 [-0.04, 0.24]
	, 	0.12 [0.05, 0.19] 0.12 [-0.07, 0.31]
		0.12 [-0.07, 0.31]
192	` ⊨= -1	0.14 [-0.00, 0.28]
4354		0.14 [0.11, 0.17]
		0.14 [0.06, 0.22] 0.15 [0.03, 0.27]
599		0.15 [0.07, 0.23]
1070		0.16 [0.10, 0.22]
351	 ■ 	0.19 [0.09, 0.30]
		0.20 [0.13, 0.27]
		0.21 [0.10, 0.33] 0.21 [0.03, 0.40]
		0.23 [0.17, 0.30]
392	H ≡ H	0.23 [0.13, 0.33]
182	⊢= +	0.23 [0.09, 0.38]
274	븝	0.23 [0.12, 0.35] 0.23 [0.18, 0.29]
		0.23 [0.18, 0.29]
415	H al	0.24 [0.15, 0.34]
81	 ■ 	0.26 [0.03, 0.48]
602	H	0.26 [0.18, 0.34]
		0.28 [0.20, 0.36] 0.29 [0.27, 0.31]
	<u>.</u>	0.29 [0.27, 0.31]
8615	-	0.34 [0.32, 0.36]
202	⊢= ⊢	0.35 [0.22, 0.49]
		0.35 [0.14, 0.57]
	F=-	0.35 [0.22, 0.49] 0.39 [0.37, 0.41]
112		0.40 [0.21, 0.59]
8615	=	0.42 [0.40, 0.44]
212	H=1	0.46 [0.32, 0.60]
		0.51 [0.43, 0.59] 0.55 [0.29, 0.81]
		0.58 [0.43, 0.72]
157		0.73 [0.57, 0.88]
253	H ≡ H	-0.07 [-0.19, 0.05]
		0.00 [-0.12, 0.12]
		0.02 [-0.10, 0.14] 0.05 [-0.06, 0.16]
275		0.06 [-0.06, 0.18]
60	⊢ •−	0.07 [-0.19, 0.33]
253	 ≡ +	0.07 [-0.05, 0.19]
		0.07 [-0.08, 0.22] 0.09 [-0.06, 0.25]
	Ĥ a	0.09 [-0.03, 0.21]
6802		0.09 [0.07, 0.11]
275	(■ 1	0.10 [-0.02, 0.22]
		0.11 [-0.01, 0.23]
		0.12 [-0.02, 0.26] 0.13 [0.02, 0.24]
263	-	0.13 [0.01, 0.25]
436	H	0.14 [0.05, 0.24]
		0.14 [0.02, 0.26]
261	?	0.14 [0.02, 0.26] 0.19 [0.14, 0.25]
	Y	0.10 [0.14, 0.20]
261) = 1	0.14 [0.02, 0.26]
275	 	0.15 [0.03, 0.27]
	} 	0.15 [0.03, 0.27] 0.15 [0.03, 0.27]
188		0.15 [0.01, 0.30]
96	⊢ •−1	0.17 [-0.03, 0.37]
436	H	0.18 [0.09, 0.28]
	H al	0.18 [0.06, 0.31] 0.20 [0.08, 0.33]
		0.20 [0.08, 0.33]
154	F = -1	0.22 [0.06, 0.38]
253	H ■ H	0.22 [0.10, 0.35]
261	 	0.24 [0.12, 0.37]
		0.24 [0.12, 0.37] 0.24 [0.12, 0.37]
253 96	⊬= -	0.24 [0.12, 0.37]
261	H = H	0.26 [0.13, 0.38]
310	H ≡ H	0.26 [0.14, 0.37]
253	⊢ ■+	0.31 [0.19, 0.43]
		0.34 [0.14, 0.55] 0.34 [0.22, 0.46]
96		0.35 [0.15, 0.56]
188	⊦ = ⊣	0.35 [0.21, 0.50]
253	H el l	0.35 [0.23, 0.48]
		0.37 [0.24, 0.49] 0.38 [0.25, 0.50]
		0.38 [0.25, 0.50]
253		0.41 [0.29, 0.54]
261	H■H	0.44 [0.31, 0.56]
261	H art	0.45 [0.33, 0.57]
253	. ■ 	0.51 [0.39, 0.63]
96	- - 	0.52 [0.32, 0.73] 0.55 [0.43, 0.67]
		0.58 [0.42, 0.74]
261 154		
	⊢• →	0.73 [0.52, 0.93]
154	⊢= →	0.73 [0.52, 0.93] 0.21 [0.16, 0.27]
154	→	0.21 [0.16, 0.27]
154	→ → →	
	30 98 1075 210 316 1142 316 320 321 320 331 486 486 282 220 2112 110 282 217 207 703 112 99 192 4354 601 282 259 90 1070 351 784 306 112 261 150 8815 597 8815 891 8815 597 597 597 598 598 598 598 598 598 598 598 598 598	98 1075

End of Code