R-Code Tutorial: Conducting a Meta-Analysis

A Step-by-Step Example to Analyzing the Effects of Sulfur Fertilization on Wheat Yield

Gustavo Roa

2024-05-21

Table of contents

1 Introduction **2**

2 Data Preparation **2**

3 Load Necessary Packages **2**

4 Load Necessary Funtions **3**
4.1 Bootstrap Meta-Analysis Model 3
4.2 Summarize Bootstraps 5

5 Load data **6**

6 Maps **7**

7 Yield **8**
7.1 Prepare and Transform Data: Calculate effect sizes & pooled sample variance . 8
7.2 Aggregate Data by Study 9
7.3 Run model - intercept only 9
7.4 Model heterogeneity, publication bias and influential cases 10
7.5 Forest plot 12
7.6 Influential Studies Diagnosis 15
7.7 Run unbootstrapped model - Texture 15
7.8 Pooled effects, intercept only model 16
7.9 Test potential moderators 17
7.10 Summarize Bootstrap Results 17
7.11 Plot 17

1 Introduction

Meta-analysis is a powerful statistical technique that allows researchers to combine and analyze data from multiple studies to obtain a more comprehensive understanding of a particular research question. In this tutorial, we will demonstrate how to conduct a meta-analysis in R, using the example of investigating the effects of sulfur fertilization on wheat yield and protein content.

2 Data Preparation

Before conducting the meta-analysis, it is essential to prepare the data in a suitable format. The data should be organized in a spreadsheet or data frame, with each row representing a single study or comparison, and columns containing relevant information such as study ID, treatment and control means, sample sizes, and measures of variability (e.g., standard deviation or standard error). For this example we will be using the file "Sulfur_datasets_GRoa.xlsx", this dataset includes the compiled data from the 55 studies used in the meta-analysis, which were sourced from published literature.

Note: Please ensure that there is a folder named "output" in the same directory where your dataset and the R code tutorial file are located. This is necessary for the code to execute properly.

3 Load Necessary Packages

To begin, we need to load the necessary R packages for data wrangling, visualization, and meta-analysis.

```
# Required packages

library(tidyverse) # data wrangling and visualization
library(readxl) # read excel files
library(bayestestR) # bootstrap summaries
library(metafor) # meta-analytic model
library(maps) # world maps
library(parallel) # parallel processing
library(multidplyr) # A Multi-Process 'dplyr'
library(purrr) # Functional Programming Tools
```

4 Load Necessary Funtions

The following functions are copied directly from the file functions_sys_reviews.R from Giordano, N. 2023. Systematic-reviews-in-R. Github

4.1 Bootstrap Meta-Analysis Model

This function performs a bootstrap meta-analysis, either with or without a moderator variable.

```
# Function for back transforming effect sizes

trans = function(x){
  out = (exp(x)-1)*100
  return(out)
}

# Bootstrap meta-analytic model

bootstrap_rma = function(data, #input data
                         response_variable,  # name of the response
 ↪  variable in between quotations
                         moderator, # name of the moderator variable if
 ↪  any just type in NA
                         boot_num, # number of  bootstrap samples
                         cores = 16, # number of cores available in your
                          ↪  computer
                         seed = 1) # seed for replicability
{
  set.seed(seed)
  cluster <- new_cluster(cores)
  cluster_library(cluster, c("metafor","dplyr", "purrr", "modelr",
   ↪  "multidplyr"))



  if (is.na(moderator)) {

    system.time(
      data %>%
        rename(.,
```

```
            RV = all_of(response_variable)) %>%
      #Run bootstrapped models
      modelr::bootstrap(n =  boot_num, id = 'boot_num') %>%
      group_by(boot_num) %>%
      multidplyr::partition(cluster) %>%
      mutate(fit = strap %>% map(~rma(yi = RV,
                                      vi = VAR,
                                      weights = W,
                                      data = .x)
      )
      ) %>%
      mutate(mod_val = fit %>% map(~data.frame(ESTIM = coef(.x),
                                               PVAL =
                                                 ↪  summary(.x)$pval)
      )
      )  %>%
      dplyr::select(-fit, -strap) %>%
      dplyr::collect() %>%
      unnest(mod_val) %>%
      saveRDS(paste0("output/",response_variable,"_mod.RData"))
  )

}

if (!is.na(moderator)) {

  system.time(
    data %>%
      rename(.,
             MOD = all_of(moderator),
             RV = all_of(response_variable)) %>%
      #Run bootstrapped models
      modelr::bootstrap(n =  boot_num, id = 'boot_num') %>%
      group_by(boot_num) %>%
      multidplyr::partition(cluster) %>%
      mutate(fit = strap %>% map(~rma(yi = RV,
                                      vi = VAR,
                                      mods = ~ 0 + MOD,
                                      weights = W,
                                      data = .x)
      )
```

```
        ) %>%
        mutate(mod_val = fit %>% map(~data.frame(MOD =
         ↪  levels(as.factor(pluck(.x, "data")$MOD)),
                                                 ESTIM = coef(.x),
                                                 PVAL =
                                                  ↪  summary(.x)$pval)
        )
        )  %>%
        dplyr::select(-fit, -strap) %>%
        dplyr::collect() %>%
        unnest(mod_val) %>%

         ↪  saveRDS(paste0("output/",response_variable,"_",moderator,"_mod.RData"))
    )
  }
}
```

4.2 Summarize Bootstraps

This function summarizes the bootstrap results, providing confidence intervals for the estimated effects.

```
# Summarise bootstraps

`%nin%` <- Negate(`%in%`)

summarise_bootstraps = function(data){  # data output as obteined from
 ↪  bootstrap_rma() function

  if ("MOD" %in% colnames(data)) {

    boot = data %>%
      group_by(MOD) %>%
      summarise_at(vars(ESTIM, PVAL), list(q975 = ~quantile(., 0.975,
       ↪  na.rm=T),
                                           q025 = ~quantile(., 0.025,
                                            ↪  na.rm=T),
```

```
                                                #hdi = ~hdi(.), you might
                                                 ↪  want to use HDI depending
                                                 ↪  on the distribution of
                                                 ↪  the resamples. HDI is
                                                 ↪  worth using it for
                                                 ↪  p-values bootstraps
                                                q500 = ~quantile(.,0.500,
                                                 ↪  na.rm=T)))

  }

  if ("MOD" %nin% colnames(data)) {

    boot = data %>%
      ungroup() %>%
      summarise_at(vars(ESTIM, PVAL), list(q975 = ~quantile(., 0.975,
       ↪  na.rm=T),
                                                q025 = ~quantile(., 0.025,
                                                 ↪  na.rm=T),
                                                #hdi = ~hdi(.), you might
                                                 ↪  want to use HDI depending
                                                 ↪  on the distribution of
                                                 ↪  the resamples. HDI is
                                                 ↪  worth using it for
                                                 ↪  p-values bootstraps
                                                q500 = ~quantile(.,0.500,
                                                 ↪  na.rm=T)))

  }


  return(boot)

}
```

5 Load data

Next, we load the dataset containing the compiled data from 55 studies.

```r
raw_data <- read_excel("Sulfur_datasets_GRoa.xlsx" ,sheet = "database")

print(raw_data)
```

```
# A tibble: 545 x 39
   study authors year  country paper_type latitude longitude site_year
   year_site
   <dbl> <chr>   <chr> <chr>   <chr>         <dbl>     <dbl> <chr>     <chr>
 1     1 Soofiz~ 2022  Italy   Peer revi~     43.3      11.8 2018-2019
 Multiple
 2     2 Kaiser~ 2019  United~ Peer revi~     46.4     -94.2 2014-2016
 Multiple
 3     2 Kaiser~ 2019  United~ Peer revi~     46.4     -94.2 2014-2016
 Multiple
 4     3 Tabak ~ 2019  Poland  Peer revi~     50.1      19.8 2015      Single
 5     3 Tabak ~ 2019  Poland  Peer revi~     50.1      19.8 2015      Single
 6     3 Tabak ~ 2019  Poland  Peer revi~     50.1      19.8 2015      Single
 7     3 Tabak ~ 2019  Poland  Peer revi~     50.1      19.8 2016      Single
 8     3 Tabak ~ 2019  Poland  Peer revi~     50.1      19.8 2016      Single
 9     3 Tabak ~ 2019  Poland  Peer revi~     50.1      19.8 2016      Single
10     3 Tabak ~ 2019  Poland  Peer revi~     50.1      19.8 2017      Single
# i 535 more rows
# i 30 more variables: site <chr>, varieties <chr>, type_wheat <chr>,
#   previous_crop <chr>, trt_name <chr>, sulfur_rate_kg_ha <chr>,
#   sulfur_fertilizer <chr>, sulfur_timing <chr>, sulfur_aplication <chr>,
#   nitrogen_rate_kg_ha <chr>, nitrogen_fertilizer <chr>,
#   nitrogen_timing <chr>, nitrogen_aplication <chr>, irrigation <chr>,
#   tillage <chr>, experimental_design <chr>, rep_or_block <dbl>, pH <dbl>,
...
```

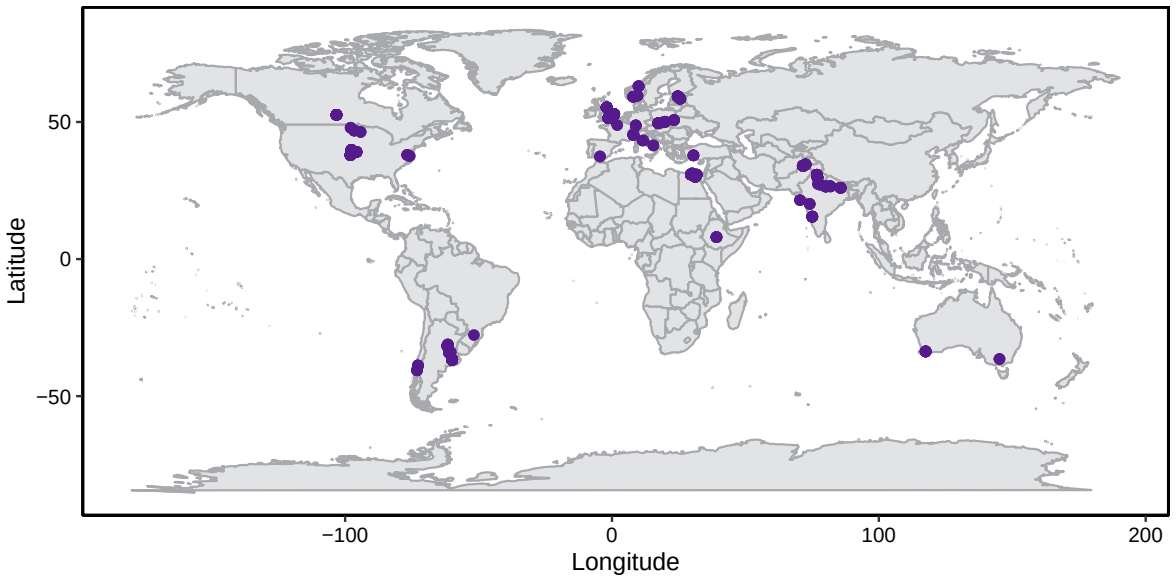
6 Maps

We can create a map to visualize the geographic distribution of the studies included in the meta-analysis.

```r
map <- ggplot(data = raw_data) +
  borders("world", colour="#a7a9ac", fill="#e2e3e4") +
  geom_point(aes(x=longitude, y=latitude), color="purple4", size=2) +
  xlab("Longitude") + ylab("Latitude") +
  theme_classic(base_size = 12) +
```

```
  theme(axis.text = element_text(color = "black"), panel.background =
   ↪  element_rect(colour = "black", linewidth=1))

map
```



7 Yield

7.1 Prepare and Transform Data: Calculate effect sizes & pooled sample variance

```
# Yield
data_all_yield =
  raw_data %>% mutate(

            # Response Ratio
            RR = log(treated_yield_kg_ha/control_yield_kg_ha),

            # Calculate pooled sampling variance

            VAR = (
             ↪  (sd_treated_yield_kg_ha^2)/(rep_or_block*(treated_yield_kg_ha^2))
             ↪  ) +
```

```
                (
  ↪ (sd_control_yield_kg_ha^2)/(rep_or_block*(control_yield_kg_ha^2)) ),

            # Weights
            W = 1/VAR
            )
```

7.2 Aggregate Data by Study

```
# Example data structure with study identifier
data_yield_study <- data_all_yield %>%
  group_by(year, authors, study) %>%
  summarize(
    yi = sum(RR * W) / sum(W),  # Weighted mean effect size
    vi = sum(VAR) / n(),        # Mean variance
    .groups = 'drop',
    N = n())
```

7.3 Run model - intercept only

```
# Conduct meta-analysis
mod_yield <- rma(yi = yi,
                 vi = vi,
                 data = data_yield_study)

print(mod_yield)
```

```
Random-Effects Model (k = 55; tau^2 estimator: REML)

tau^2 (estimated amount of total heterogeneity): 0.0023 (SE = 0.0008)
tau (square root of estimated tau^2 value):      0.0478
I^2 (total heterogeneity / total variability):   74.13%
H^2 (total variability / sampling variability):  3.87

Test for Heterogeneity:
Q(df = 54) = 137.2079, p-val < .0001
```

```
Model Results:

estimate      se    zval    pval   ci.lb   ci.ub
  0.0414  0.0094  4.3819  <.0001  0.0229  0.0599  ***

---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Transform to percentage change

intrcpt <- (exp(coef(mod_yield)) - 1) * 100

print(intrcpt)
```

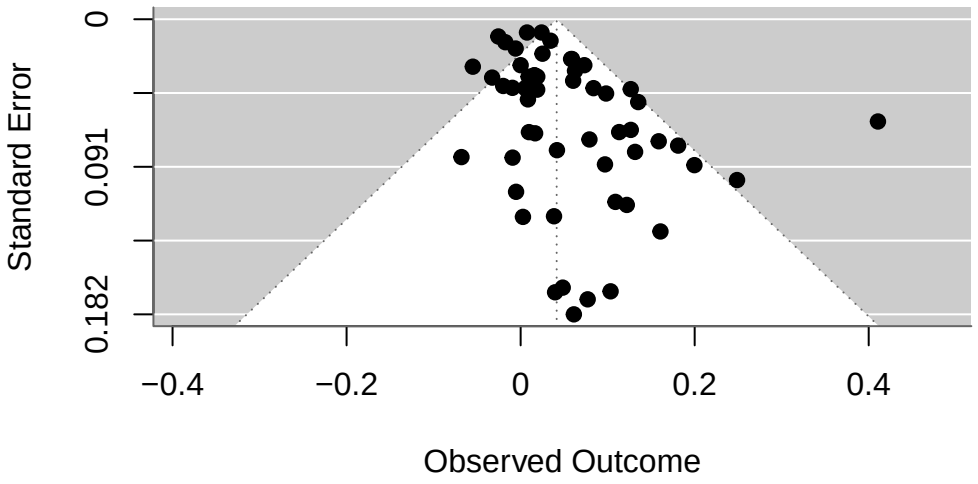
```
 intrcpt
4.227655
```

7.4 Model heterogeneity, publication bias and influential cases

```r
# Assess heterogeneity
print(mod_yield$I2)
```

```
[1] 74.13303
```

```r
# Publication bias analysis
funnel(mod_yield)
```




```
regtest(mod_yield)
```

```
Regression Test for Funnel Plot Asymmetry

Model:     mixed-effects meta-regression model
Predictor: standard error

Test for Funnel Plot Asymmetry: z =  3.5480, p = 0.0004
Limit Estimate (as sei -> 0):   b = -0.0009 (CI: -0.0263, 0.0245)
```

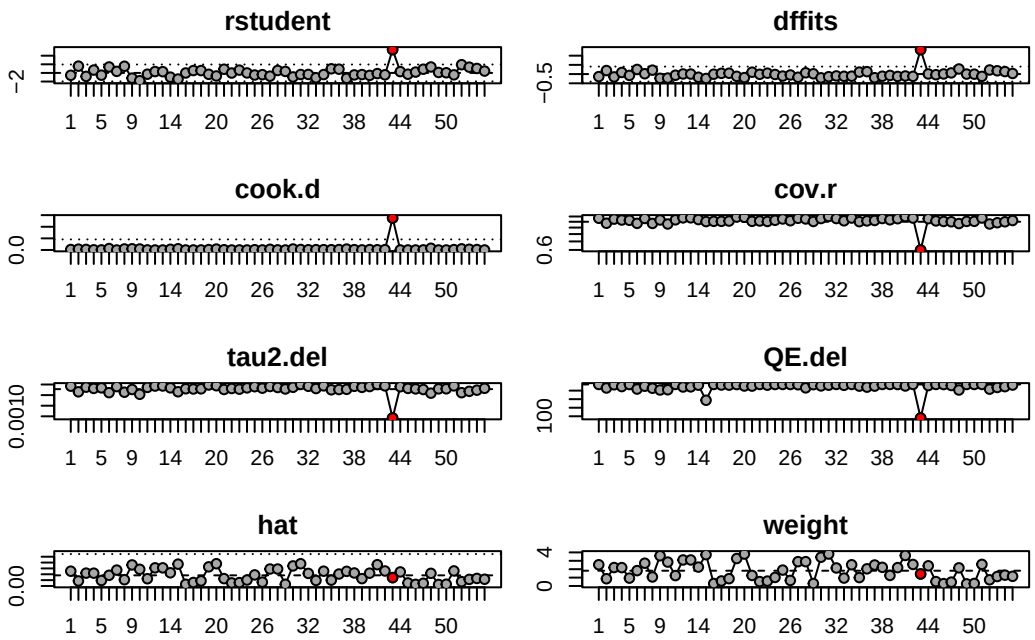
```
ranktest(mod_yield)
```

```
Warning in cor.test.default(yi.star, vi, method = "kendall", exact = exact):
Cannot compute exact p-value with ties


Rank Correlation Test for Funnel Plot Asymmetry

Kendall's tau = 0.1496, p = 0.1070
```

```r
# Diagnostics for potential outliers and influential cases
inf <- influence(mod_yield)
plot(inf)
```



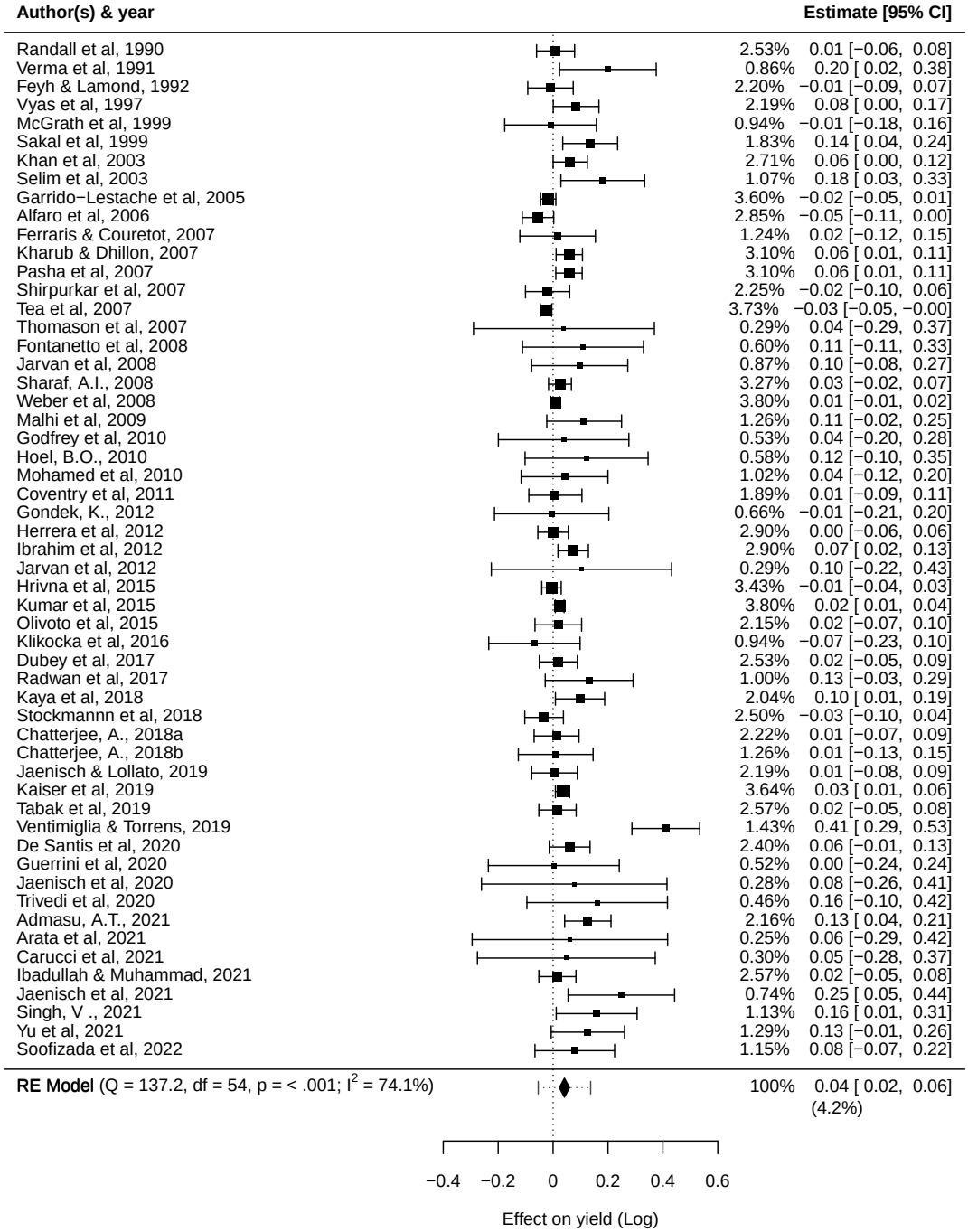
7.5 Forest plot

```r
forest(mod_yield, xlim=c(-2,1.5), showweights=TRUE,
       xlab= "Effect on yield (Log)",
       cex=.8, slab=paste(authors, year, sep=", "), addcred = TRUE,
        ↪  header="Author(s) & year", col = "black")
op <- par(cex=.8, font=2)

par(op)

text(-2, -1, pos=4, cex=0.8, bquote(paste("RE Model (Q = ",
     .(formatC(mod_yield$QE, digits=1, format="f")), ", df = ",
 ↪   .(mod_yield$k - mod_yield$p),
     ", p = ", .(formatC("< .001", format="f")), "; ", I^2, " = ",
     .(formatC(mod_yield$I2, digits=1, format="f")), "%)")))
```

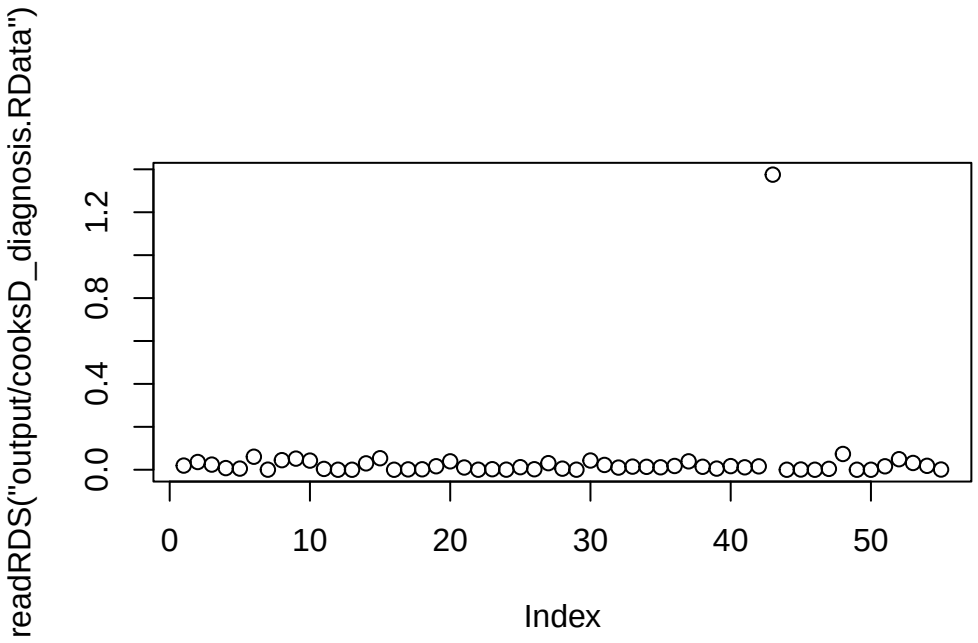
```
text(0.894, -2.3, pos=4, cex=0.8, bquote(paste("(", .(formatC(intrcpt,
 ↪  digits=1, format="f")), "%)")))
```


Author(s) & year
Estimate [95% CI]
Randall et al, 1990 2.53% 0.01 [−0.06, 0.08]
Verma et al, 1991 0.86% 0.20 [0.02, 0.38]
Feyh & Lamond, 1992 2.20% −0.01 [−0.09, 0.07]
Vyas et al, 1997 2.19% 0.08 [0.00, 0.17]
McGrath et al, 1999 0.94% −0.01 [−0.18, 0.16]
Sakal et al, 1999 1.83% 0.14 [0.04, 0.24]
Khan et al, 2003 2.71% 0.06 [0.00, 0.12]
Selim et al, 2003 1.07% 0.18 [0.03, 0.33]
Garrido−Lestache et al, 2005 3.60% −0.02 [−0.05, 0.01]
Alfaro et al, 2006 2.85% −0.05 [−0.11, 0.00]
Ferraris & Couretot, 2007 1.24% 0.02 [−0.12, 0.15]
Kharub & Dhillon, 2007 3.10% 0.06 [0.01, 0.11]
Pasha et al, 2007 3.10% 0.06 [0.01, 0.11]
Shirpurkar et al, 2007 2.25% −0.02 [−0.10, 0.06]
Tea et al, 2007 3.73% −0.03 [−0.05, −0.00]
Thomason et al, 2007 0.29% 0.04 [−0.29, 0.37]
Fontanetto et al, 2008 0.60% 0.11 [−0.11, 0.33]
Jarvan et al, 2008 0.87% 0.10 [−0.08, 0.27]
Sharaf, A.I., 2008 3.27% 0.03 [−0.02, 0.07]
Weber et al, 2008 3.80% 0.01 [−0.01, 0.02]
Malhi et al, 2009 1.26% 0.11 [−0.02, 0.25]
Godfrey et al, 2010 0.53% 0.04 [−0.20, 0.28]
Hoel, B.O., 2010 0.58% 0.12 [−0.10, 0.35]
Mohamed et al, 2010 1.02% 0.04 [−0.12, 0.20]
Coventry et al, 2011 1.89% 0.01 [−0.09, 0.11]
Gondek, K., 2012 0.66% −0.01 [−0.21, 0.20]
Herrera et al, 2012 2.90% 0.00 [−0.06, 0.06]
Ibrahim et al, 2012 2.90% 0.07 [0.02, 0.13]
Jarvan et al, 2012 0.29% 0.10 [−0.22, 0.43]
Hrivna et al, 2015 3.43% −0.01 [−0.04, 0.03]
Kumar et al, 2015 3.80% 0.02 [0.01, 0.04]
Olivoto et al, 2015 2.15% 0.02 [−0.07, 0.10]
Klikocka et al, 2016 0.94% −0.07 [−0.23, 0.10]
Dubey et al, 2017 2.53% 0.02 [−0.05, 0.09]
Radwan et al, 2017 1.00% 0.13 [−0.03, 0.29]
Kaya et al, 2018 2.04% 0.10 [0.01, 0.19]
Stockmannn et al, 2018 2.50% −0.03 [−0.10, 0.04]
Chatterjee, A., 2018a 2.22% 0.01 [−0.07, 0.09]
Chatterjee, A., 2018b 1.26% 0.01 [−0.13, 0.15]
Jaenisch & Lollato, 2019 2.19% 0.01 [−0.08, 0.09]
Kaiser et al, 2019 3.64% 0.03 [0.01, 0.06]
Tabak et al, 2019 2.57% 0.02 [−0.05, 0.08]
Ventimiglia & Torrens, 2019 1.43% 0.41 [0.29, 0.53]
De Santis et al, 2020 2.40% 0.06 [−0.01, 0.13]
Guerrini et al, 2020 0.52% 0.00 [−0.24, 0.24]
Jaenisch et al, 2020 0.28% 0.08 [−0.26, 0.41]
Trivedi et al, 2020 0.46% 0.16 [−0.10, 0.42]
Admasu, A.T., 2021 2.16% 0.13 [0.04, 0.21]
Arata et al, 2021 0.25% 0.06 [−0.29, 0.42]
Carucci et al, 2021 0.30% 0.05 [−0.28, 0.37]
Ibadullah & Muhammad, 2021 2.57% 0.02 [−0.05, 0.08]
Jaenisch et al, 2021 0.74% 0.25 [0.05, 0.44]
Singh, V ., 2021 1.13% 0.16 [0.01, 0.31]
Yu et al, 2021 1.29% 0.13 [−0.01, 0.26]
Soofizada et al, 2022 1.15% 0.08 [−0.07, 0.22]
RE Model (Q = 137.2, df = 54, p = < .001; I² = 74.1%) 100% 0.04 [0.02, 0.06]
(4.2%)
−0.4 −0.2 0 0.2 0.4 0.6
Effect on yield (Log)

7.6 Influential Studies Diagnosis

```
 cooks.distance.rma.uni(model = mod_yield, progbar = T) %>%
   saveRDS("output/cooksD_diagnosis.RData")

plot(readRDS("output/cooksD_diagnosis.RData"))
```




```
influential_cases = c(2, 3, 4)

data.imp_es_ic =
  data_all_yield %>%
  mutate(W = case_when(study %in% influential_cases ~ 0, T~W))
```

7.7 Run unbootstrapped model - Texture

The weight of influential studies is set to zero. This step involves conducting a mixed-effects meta-regression model. Using the texture_group.

```r
mod_texture = rma(yi = RR,
          vi = VAR,
          weights = W,
          mods = ~ 0 + texture_group, # change
          data = data.imp_es_ic)
```

```
Warning: 163 studies with NAs omitted from model fitting.
```

```r
print(mod_texture)
```

```
Mixed-Effects Model (k = 382; tau^2 estimator: REML)

tau^2 (estimated amount of residual heterogeneity):     0.0062 (SE = 0.0006)
tau (square root of estimated tau^2 value):             0.0787
I^2 (residual heterogeneity / unaccounted variability): 83.31%
H^2 (unaccounted variability / sampling variability):   5.99

Test for Residual Heterogeneity:
QE(df = 378) = 1417.0302, p-val < .0001

Test of Moderators (coefficients 1:4):
QM(df = 4) = 8.7406, p-val = 0.0679

Model Results:

                    estimate      se     zval    pval    ci.lb   ci.ub
texture_groupClay     0.0092  0.0148   0.6244  0.5324  -0.0198  0.0383
texture_groupLoam    -0.0048  0.0136  -0.3501  0.7263  -0.0315  0.0219
texture_groupSandy    0.0290  0.0102   2.8537  0.0043   0.0091  0.0490  **
texture_groupSilty    0.0067  0.0228   0.2912  0.7709  -0.0381  0.0514

---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

7.8 Pooled effects, intercept only model

Bootstrap analysis with at least 1000 bootstrap samples is recommended. Ensure the number of cores available on your computer is used for parallel processing. This may take several minutes to run.

```r
ncores <- detectCores()

bootstrap_rma(data = data.imp_es_ic, response_variable = "RR", moderator
 ↪  = NA, boot_num = 1000, cores = ncores)
```

7.9 Test potential moderators

This section tests the effects of potential moderators, in this case, the "texture_group."

```r
bootstrap_rma(data = data.imp_es_ic, response_variable = "RR",moderator
 ↪  = "texture_group", boot_num = 1000, cores = ncores)
```

```
   user  system elapsed
   0.77    0.25  112.53
```

7.10 Summarize Bootstrap Results

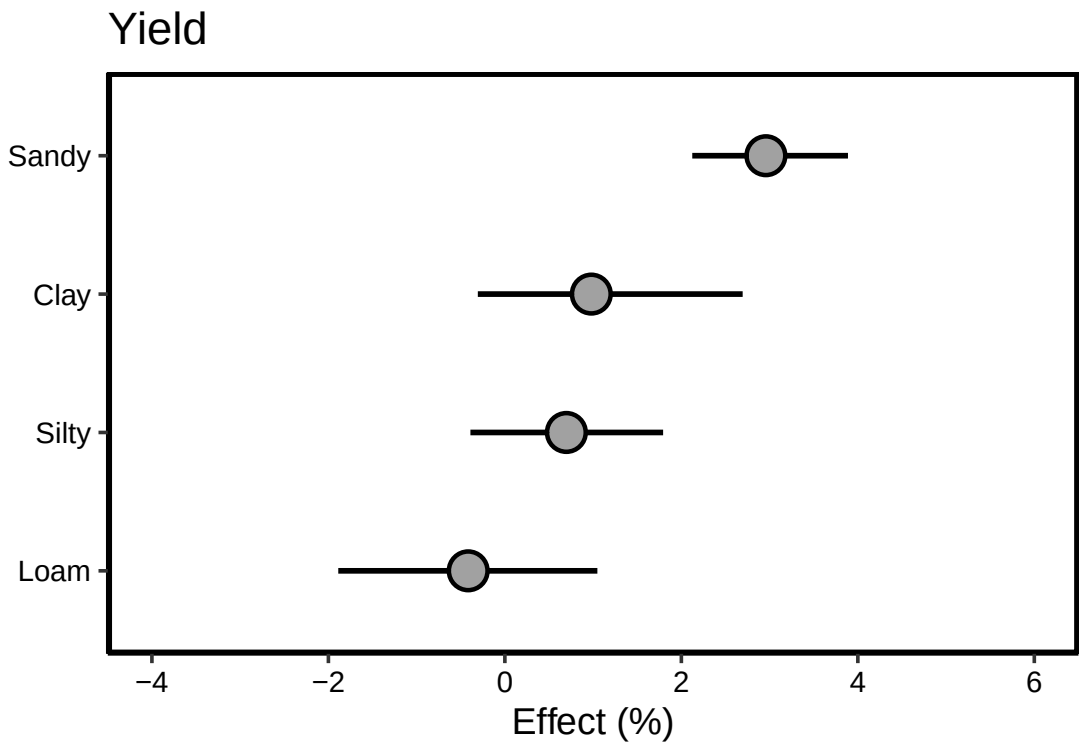
```r
df.plot =
 ↪  summarise_bootstraps(readRDS("output/RR_texture_group_mod.RData"))

df.plot$MOD <- factor(df.plot$MOD, levels=c("Loam","Silty", "Clay",
 ↪  "Sandy"))

df.plot
```

```
# A tibble: 4 x 7
  MOD   ESTIM_q975 PVAL_q975 ESTIM_q025 PVAL_q025 ESTIM_q500 PVAL_q500
  <fct>      <dbl>     <dbl>      <dbl>     <dbl>      <dbl>     <dbl>
1 Clay      0.0266    0.969    -0.00307  0.0634      0.00975   0.512
2 Loam      0.0104    0.982    -0.0191   0.140      -0.00416   0.656
3 Sandy     0.0381    0.0616    0.0210   0.000127    0.0292    0.00410
4 Silty     0.0178    0.979    -0.00391  0.298       0.00695   0.745
```

7.11 Plot

Generate a plot to visualize the bootstrap results, showing the effect sizes for different texture groups.

```r
FigX <- df.plot %>%
  ggplot()+
  geom_linerange(aes(xmin = trans(ESTIM_q975), xmax = trans(ESTIM_q025),
   ↪  y = MOD), linewidth = 1)+
  geom_point(aes(y = MOD, x = trans(ESTIM_q500)), shape = 21, size = 6,
   ↪  stroke = 1.2, fill = "#A1A1A1", color = "black") +
  xlim(-4,6)+
  labs(y = "", x = "Effect (%)") +
  theme_classic(base_size = 14) +
  theme(axis.text = element_text(color = "black")) +
  theme(panel.background = element_rect(colour = "black",
   ↪  linewidth=1.5)) +
guides(fill = "none") +
  ggtitle("Yield")

FigX
```