Meta-Analysis on Digital Cognitive Training

Table of Contents

# 1. Introduction

This R Markdown report contains the entire script used for the meta-analysis on digital cognitive training for patients with alcohol use disorder.

It is build as an R Environment (renv) for managing R package versions and maximising the reproducibility for future analyses, and all package versions are stored in the renv.lock-file.

The packages needed for this report are knitr, osfr, devtools, ggplot, robvis, readxl, openxlsx, dplyr, meta, grid, and grateful .The knitr package is used for knitting the R markdown file to PDF or word. The next three packages are used to create a traffic plot and a summary of the risk of bias ratings. The robvis package-version required, is the latest developer-version, as this will display study labels horizontally. The readxl and openxlsx packages are used for reading and opening the Excel dataset, whereas the dplyr package is used for transforming and preparing the data for the analyses. All the effect sizes are calculated using the meta package, and forest and funnel plots are generated using this package as well. The grid package, is used to add additional statistics to the plots that are not standard for the meta package. Finally, to document all the packages used and the corresponding versions, we used the grateful package to create a summary and a list of references for each of the used packages in this analysis report.

The full R analysis environment, including all scripts and supporting files, is available in the following public GitHub repository: <https://github.com/mrst3rz/digital-ct-for-aud>

## 1.1 Loading and installing the packages

To prepare the R environment for the analyses, we first need to install the required packages.

# List of required CRAN packages  
packages\_required <- c("osfr", "rmarkdown", "knitr", "readxl", "openxlsx", "dplyr", "meta",   
 "grid", "devtools", "tidyr", "tidyverse", "ggplot2", "grateful")  
  
# Function to install missing CRAN packages  
install\_if\_missing <- function(pkg) {  
 if (!requireNamespace(pkg, quietly = TRUE)) {  
 install.packages(pkg)  
 }  
}  
  
# Install all missing CRAN packages  
invisible(sapply(packages\_required, install\_if\_missing))  
  
# Load all the packages  
invisible(lapply(packages\_required, library, character.only = TRUE))  
  
# Install robvis from GitHub to display study labels horizontally  
if (!requireNamespace("robvis", quietly = TRUE)) {  
 devtools::install\_github("mcguinlu/robvis")  
}  
  
# Load latest robvis-version  
library(robvis)

A snapshot of the R environment can be performed, after the packages have been installed by running renv::snapshot.

First we load the project data from the online repository OSF, which is done with the package osfr. This can be skipped, if the data has already been stored locally. A token for accessing the project data is needed during the embargo period.

# Get the OSF project node  
osf\_proj <- osf\_retrieve\_node("3s2mr")  
  
# List all folders inside the project  
osf\_folders <- osf\_ls\_files(osf\_proj, type = "folder")  
  
# First, get the folder named "Risk of bias data"  
osf\_rob\_folder <- osf\_folders %>%  
 filter(name == "Risk of bias data") %>%  
 pull(id)  
  
# Then, get the folder named "Clinical outcome data"  
osf\_clinical\_folder <- osf\_folders %>%  
 filter(name == "Clinical outcome data") %>%  
 pull(id)  
  
# Retrieve both folders using osf\_retrieve\_file  
osf\_rob\_folder <- osf\_retrieve\_file(osf\_rob\_folder)  
osf\_clinical\_folder <- osf\_retrieve\_file(osf\_clinical\_folder)  
  
# List files inside the folders  
osf\_rob\_files <- osf\_ls\_files(osf\_rob\_folder)  
osf\_clinical\_files <- osf\_ls\_files(osf\_clinical\_folder)  
  
# Filter and download the desired Excel file on risk of bias data  
osf\_rob\_files <- osf\_rob\_files %>%  
 filter(name == "Risk of Bias Data.csv") %>%  
 osf\_download(path = tempdir(), conflicts = "overwrite")  
  
# Filter and download the desired Excel file on clinical outcomes  
osf\_clinical\_files <- osf\_clinical\_files %>%  
 filter(name == "Extracted Clinical Outcome Data.xlsx") %>%  
 osf\_download(path = tempdir(), conflicts = "overwrite")

## 1.3 Importing risk of bias data

Before conducting the meta-analyses, we want to create two plots for the risk of bias (RoB) assessments for the included studies. We used the revised version of the risk of bias tool (RoB 2), and the plots adhere to the domains specified by Sterne et al. (2019).

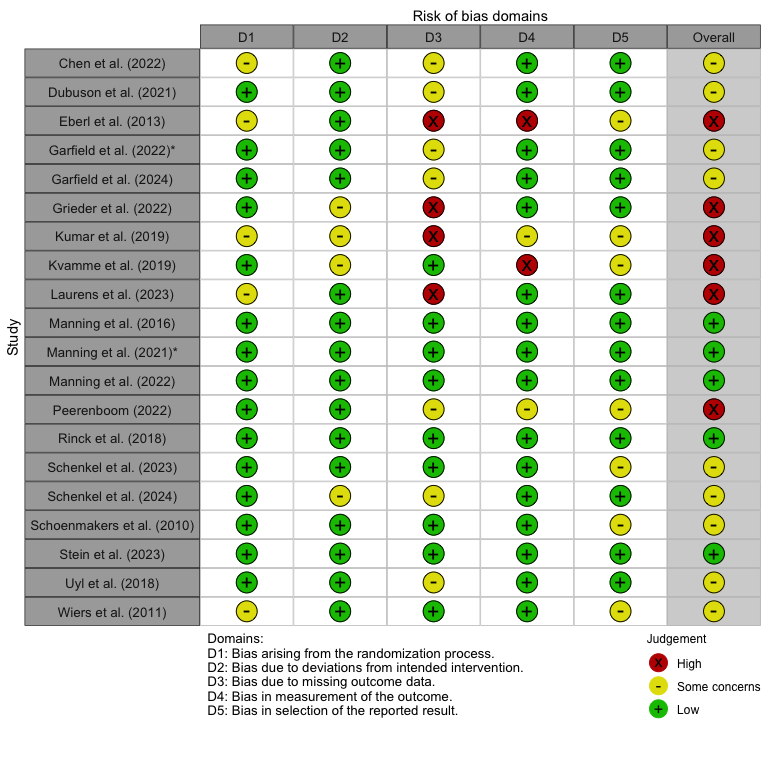
The first step to display the risk of bias results, requires that a separate data set is imported, and as this file is formatted as a csv-file, we can use the readr-package.

# Importing csv-file containing the RoB data set  
rob\_d <- read.csv(osf\_rob\_files$local\_path, sep = ";", header = TRUE)

## 1.4 Displaying the risk of bias

After the data set has been imported, we can create the traffic light plot using the function called rob\_traffic\_light(), which will display the individual ratings for each domain for each of the included studies.

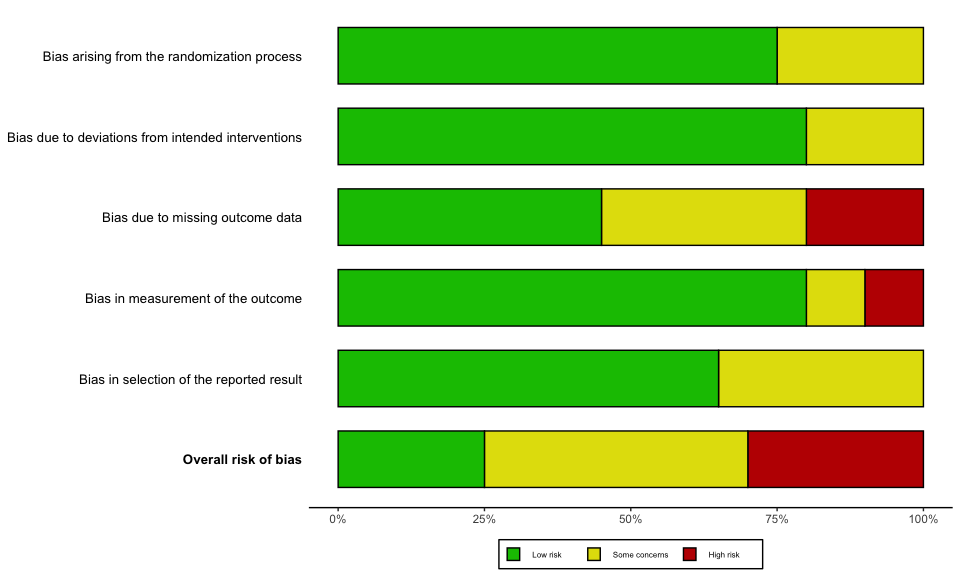
# Create a traffic plot displaying individual bias ratings  
rob\_traffic\_light(rob\_d, tool = "ROB2", psize = 7)



1. Traffic plot of individual risk of bias ratings

To display the summary of risk of bias across the domains including the overall risk of bias, we can create a new bar plot using the rob\_summary() function. To begin with we set the weighted parameter to FALSE, as we will calculate the weights at the end of the report.

# Create a summary (barplot) of risk of bias assessments  
rob\_summary(rob\_d, tool = "ROB2", weighted = FALSE) # weighted barplot will be created later



1. Summary of risk of bias ratings

# 2. Primary analyses on short-term effects (0-3 months)

Now that the required libraries have been installed and loaded, we can initiate the primary analyses. For this first part of our primary analysis, we import the excel-dataset “Extracted Clinical Outcome Data”.

## 2.1 Short-term effects on continuous abstinence

As we are starting with our primary outcome of continuous abstinence (CA), we only need to import the first sheet of the dataset named “Abstinence”. These analyses will also be performed on the short-term effects, which means that we will need to filter the time point to “0-3 months”.

# Import data on abstinence as "ca\_d" and preserve original order of studies  
ca\_d <- read\_excel(osf\_clinical\_files$local\_path, sheet = "Abstinence") %>%  
 mutate(row\_id = row\_number())

### 2.1.1 Conversion of continuous and categorical variables

We have identified two studies Grieder et al. (2022) and Stein et al. (2023), who both reported percentage of days abstinent (PDA), which we can include in our analyses by converting the continuous outcomes to the natural logarithm of the odds ratio (lnOR) using the formula proposed by Chinn (2000):

Furthermore, we are able to convert our dichotomous outcomes on CA, when we have the lnOR, the Chinn-formula can be used to convert these values to SMDs:

These two functions can be defined for easier conversions of the effect sizes as we proceed with the analyses, and for simplicity we define their respective factors (Chinn-factor).

# Chinn-factor for converting standard error (SE) of SMD to SE of lnOR  
chinn\_smd\_to\_lnor <- (pi / sqrt (3))  
  
# Chinn-factor for converting SE of lnOR to SE of SMD  
chinn\_lnor\_to\_smd <- (sqrt(3) / pi)

### 2.1.2 Calculation of effect sizes for continuous outcomes

As the primary analyses only include the outcomes reported after 0-3 months, we first need to filter the dataset and store it as a new subset.

# Filter 0–3 month data first and store it as a new subset  
ca\_d\_3m <- ca\_d %>% filter(time\_point == "0-3 months")

The calculation of effect sizes will be done by the inbuilt functions in the meta package (Balduzzi, Rücker, and Schwarzer 2019), so for the continuous variables, we will use the metacont() function, whereas for the categorical variables, we will use the metabin() function. As these two functions need to be run separately, we will need to filter the studies, who report on PDA, and define it as a subset of our data. The studies who do not report on PDA and only dichotomous outcomes, are also filtered and stored as a subset. This can be achieved by filtering studies, which have no empty cells for the “e\_abs\_trans\_mean” columns, and for storing categorical variables, we can impose the opposite filtering rule.

# Filter studies with PDA and store it as a new dataset  
ca\_d\_cont\_3m <- ca\_d\_3m %>% filter(!is.na(e\_abs\_trans\_mean))  
  
# Filter studies only with dichotomous outcomes and store it as a new dataset  
ca\_d\_cat\_3m <- ca\_d\_3m %>% filter(is.na(e\_abs\_trans\_mean))

Now it is possible to calculate the standard mean differences (SMDs) for the studies reporting PDA by using metacont(). We apply the small study correction using Hedge’s *g*, as the study by Grieder et al. (2022) included a small sample size. Furthermore, we apply the restricted maximum likelihood (REML) and random modelling will be used for our meta-analysis.

# Calculating the SMDs (Hedge's g) using metacont()  
ca\_metacont\_3m <- metacont(  
 n.e = e\_n,  
 mean.e = e\_abs\_trans\_mean,  
 sd.e = e\_abs\_trans\_sd,  
 n.c = c\_n,  
 mean.c = c\_abs\_trans\_mean,  
 sd.c = c\_abs\_trans\_sd,  
 studlab = study,  
 data = ca\_d\_cont\_3m,  
 sm = "SMD",  
 method.smd = "Hedges",  
 method.tau = "REML",  
 method.random.ci = "classic",  
 random = TRUE  
)

After we have calculated the SMDs, we can create a new dataset or data frame and store these values alongside with the other relevant variables (e.g., time point, number of abstinents, etc.) from our original dataset. We will extract the SMD, the standard error (SE) and the 95% confidence interval (CI).

# Combine the SMDs calculated with original variables and store it in a new data frame  
ca\_d\_sumcont\_3m <- data.frame(  
 row\_id = ca\_d\_cont\_3m$row\_id,  
 study = ca\_metacont\_3m$studlab,  
 time\_point = ca\_d\_cont\_3m$time\_point,  
 super\_domain = ca\_d\_cont\_3m$super\_domain,  
 cog\_domain = ca\_d\_cont\_3m$cog\_domain,  
 total\_n = ca\_d\_cont\_3m$e\_n + ca\_d\_cont\_3m$c\_n, # Calculating total number of participants  
 c\_n = ca\_d\_cont\_3m$c\_n,  
 c\_abs = NA,  
 c\_non\_abs = NA,  
 e\_n = ca\_d\_cont\_3m$e\_n,  
 e\_abs = NA,  
 e\_non\_abs = NA,  
 # Adding calculated SMDs to the dataframe   
 SMD = ca\_metacont\_3m$TE,  
 SE = ca\_metacont\_3m$seTE,  
 LCI = ca\_metacont\_3m$lower,  
 UCI = ca\_metacont\_3m$upper,  
 check.names = FALSE # Removing the period as a space  
)

The SMDs for the two studies can now be converted to lnOR using the pre-defined Chinn-factor, and alongside them, we can exponentiate these values to obtain the ORs. Lastly, we calculate the z- and p-values and store them in the final data frame for the two studies.

# Fetching the previously stored data frame and adding the lnORs, their SE and 95% CI  
ca\_d\_sumcont\_3m <- ca\_d\_sumcont\_3m %>%   
 data.frame(  
 lnOR = ca\_metacont\_3m$TE \* chinn\_smd\_to\_lnor,  
 SElnOR = ca\_metacont\_3m$seTE \* chinn\_smd\_to\_lnor,  
 lnLCI = ca\_metacont\_3m$lower \* chinn\_smd\_to\_lnor,  
 lnUCI = ca\_metacont\_3m$upper \* chinn\_smd\_to\_lnor,  
 # Saving the ORs and its corresponding SE and 95% CI  
 OR = exp(ca\_metacont\_3m$TE \* chinn\_smd\_to\_lnor),  
 OrSE = exp(ca\_metacont\_3m$TE \* chinn\_smd\_to\_lnor) \* (ca\_metacont\_3m$seTE \* chinn\_smd\_to\_lnor),  
 OrLCI = exp(ca\_metacont\_3m$lower \* chinn\_smd\_to\_lnor),  
 OrUCI = exp(ca\_metacont\_3m$upper \* chinn\_smd\_to\_lnor),  
 # Calculating the z- and p-values  
 `z-value` = (ca\_metacont\_3m$TE \* chinn\_smd\_to\_lnor) / (ca\_metacont\_3m$seTE \* chinn\_smd\_to\_lnor  
 ),  
 `p-value` = 2 \* (1 - pnorm(abs((ca\_metacont\_3m$TE \* chinn\_smd\_to\_lnor) / (ca\_metacont\_3m$seTE \* chinn\_smd\_to\_lnor  
 )))),  
 # Adding a note to specify how the data was calculated  
 Notes = "Categorical variables converted from continuous outcome (metacont) using Chinn's formula",  
 check.names = FALSE  
)

### 2.1.3 Calculation of effect sizes for categorical outcomes

The next step is to repeat the above procedure, but this time on our categorical data using metabin().

# Calculating the lnORs using metabin()  
ca\_metabin\_3m <- metabin(  
 event.e = e\_abs,  
 n.e = e\_n,  
 event.c = c\_abs,  
 n.c = c\_n,  
 studlab = study,  
 data = ca\_d\_cat\_3m,  
 sm = "OR",  
 method.tau = "REML",  
 method.random.ci = "classic",  
 random = TRUE,  
 incr = 0.5,  
 allstudies = TRUE  
)

After we have calculated the lnORs, we can create a new data frame and store these values alongside with the other relevant variables (e.g., time point, number of abstinents, etc.) from our original dataset. As performed for the continuous outcomes, we calculate the ORs and SMDs using the Chinn-factor. Finally, we add the z- and p-values to the data-frame.

# Fetching the previously stored data frame and adding the lnORs, their SE and 95% CI  
ca\_d\_sumbin\_3m <- data.frame(  
 row\_id = ca\_d\_cat\_3m$row\_id,  
 study = ca\_metabin\_3m$studlab,  
 time\_point = ca\_d\_cat\_3m$time\_point,  
 super\_domain = ca\_d\_cat\_3m$super\_domain,  
 cog\_domain = ca\_d\_cat\_3m$cog\_domain,  
 total\_n = ca\_d\_cat\_3m$e\_n + ca\_d\_cat\_3m$c\_n,  
 c\_n = ca\_d\_cat\_3m$c\_n,  
 c\_abs = ca\_d\_cat\_3m$c\_abs,  
 c\_non\_abs = ca\_d\_cat\_3m$c\_n - ca\_d\_cat\_3m$c\_abs,  
 e\_n = ca\_d\_cat\_3m$e\_n,  
 e\_abs = ca\_d\_cat\_3m$e\_abs,  
 e\_non\_abs = ca\_d\_cat\_3m$e\_n - ca\_d\_cat\_3m$e\_abs,  
# Extracting the lnORs from the metabin results  
 lnOR = ca\_metabin\_3m$TE,  
 SElnOR = ca\_metabin\_3m$seTE,  
 lnLCI = ca\_metabin\_3m$lower,  
 lnUCI = ca\_metabin\_3m$upper,  
# Calculating the ORs from the metabin results  
 OR = exp(ca\_metabin\_3m$TE),  
 OrSE = exp(ca\_metabin\_3m$TE) \* ca\_metabin\_3m$seTE,  
 OrLCI = exp(ca\_metabin\_3m$lower),  
 OrUCI = exp(ca\_metabin\_3m$upper),  
# Caculating the SMD, SE and 95% CI using the Chinn-function  
 SMD = ca\_metabin\_3m$TE \* chinn\_lnor\_to\_smd,  
 SE = ca\_metabin\_3m$seTE \* chinn\_lnor\_to\_smd,  
 LCI = ca\_metabin\_3m$lower \* chinn\_lnor\_to\_smd,  
 UCI = ca\_metabin\_3m$upper \* chinn\_lnor\_to\_smd,  
# Adding the z- and p-values  
 `z-value` = ca\_metabin\_3m$TE / ca\_metabin\_3m$seTE,  
 `p-value` = 2 \* (1 - pnorm(abs(ca\_metabin\_3m$TE / ca\_metabin\_3m$seTE))),  
# Adding a final note on how data was converted   
 Notes = "Continuous variables converted from categorical outcome (metabin) using Chinn's formula",  
 check.names = FALSE  
)

Now that the categorical data has been stored in a separate data frame, we can combine it with the data frame for continuous data for so it can be used for our meta-analysis.

# Combine the two data frames for CA and round all the values 5 decimals  
ca\_d\_full\_3m <- bind\_rows(ca\_d\_sumcont\_3m, ca\_d\_sumbin\_3m) %>% mutate(across(where(is.numeric), ~ round(.x, 5))) %>%  
 # Arrange the studies by their pre-defined row-numbers  
 arrange(row\_id)

### 2.1.4 Meta-analysis on continuous abstinence

The short-term effects (0-3 months) of cognitive training on CA has now been calculated, and the studies with missing categorical outcomes have been converted, so our primary analyses now includes 11 studies. Before we can proceed with creating a forest plot, we need to rename the study IDs, which only purpose was differentiate different outcomes at across various time points in each of the studies. We start by defining new study labels.

# Creating study labels in accordance with APA  
study\_labels <- c(  
 "1\_chen\_2022" = "Chen et al. (2022)",  
 "2a\_uyl\_2018" = "Uyl et al. (2018)",  
 "2b\_uyl\_2018" = "Uyl et al. (2018)",  
 "3a\_dubuson\_2021" = "Dubuson et al. (2021)",  
 "3c\_dubuson\_2021" = "Dubuson et al. (2021)",  
 "3e\_dubuson\_2021" = "Dubuson et al. (2021)",  
 "3d\_dubuson\_2021" = "Dubuson et al. (2021)",  
 "4\_eberl\_2013" = "Eberl et al. (2013)",  
 "5b\_manning\_2022" = "Manning et al. (2022)",  
 "5c\_manning\_2022" = "Manning et al. (2022)",  
 "5d\_manning\_2022" = "Manning et al. (2022)",  
 "5f\_garfield\_2022" = "Garfield et al. (2022)",  
 "5g\_garfield\_2022" = "Garfield et al. (2022)",  
 "5h\_garfield\_2022" = "Garfield et al. (2022)",  
 "6\_grieder\_2022" = "Grieder et al. (2022)",  
 "7c\_garfield\_2024" = "Garfield et al. (2024)",  
 "8b\_kvamme\_2019" = "Kvamme et al. (2019)",  
 "9b\_kumar\_2019" = "Kumar et al. (2019)",  
 "9c\_kumar\_2019" = "Kumar et al. (2019)",  
 "10b\_laurens\_2023" = "Laurens et al. (2023)",  
 "10c\_laurens\_2023" = "Laurens et al. (2023)",  
 "11\_manning\_2016" = "Manning et al. (2016)",  
 "12a\_peerenboom\_2022" = "Peerenboom (2022)",  
 "12b\_peerenboom\_2022" = "Peerenboom (2022)",  
 "13a\_rinck\_2018" = "Rinck et al. (2018)",  
 "14a\_schenkel\_2023" = "Schenkel et al. (2023)",  
 "14b\_schenkel\_2023" = "Schenkel et al. (2023)",  
 "15a\_schenkel\_2024" = "Schenkel et al. (2024)",  
 "15b\_schenkel\_2024" = "Schenkel et al. (2024)",  
 "15c\_schenkel\_2024" = "Schenkel et al. (2024)",  
 "16a\_schoenmakers\_2010" = "Schoenmakers et al. (2010)",  
 "16b\_schoenmakers\_2010" = "Schoenmakers et al. (2010)",  
 "17b\_stein\_2023" = "Stein et al. (2023)",  
 "18a\_wiers\_2011" = "Wiers et al. (2011)",  
 "18b\_wiers\_2011" = "Wiers et al. (2011)"  
)

After these have been defined as values, we can assign them to our data frame on short-term effects on CA.

# Assign the new study labels to the combined data frame  
ca\_d\_full\_3m$study <- ifelse(ca\_d\_full\_3m$study %in% names(study\_labels),  
 study\_labels[ca\_d\_full\_3m$study],  
 ca\_d\_full\_3m$study)

We want to have the author labels to be sorted alphabetically rather than in accordance with the pre-specified study IDs and row numbers, so we first arrange the list of studies.

# Arrange studies alphabetically  
ca\_d\_full\_3m <- ca\_d\_full\_3m %>%  
 arrange(desc(study == "Dubuson et al. (2021)"),  
 desc(study == "Grieder et al. (2022)"),  
 desc(study == "Kvamme et al. (2019)"),  
 desc(study == "Manning et al. (2016)"),  
 desc(study == "Manning et al. (2022)"),  
 desc(study == "Peerenboom (2022)"),  
 desc(study == "Schenkel et al. (2023)"),  
 desc(study == "Schenkel et al. (2024)"),  
 desc(study == "Schoenmakers et al. (2010)"),  
 desc(study == "Stein et al. (2023)")  
 )

As we want to display number of events and total number of participants in each group for the individual studies in the forest plot, we need to prepare data frame with missing event data (i.e., studies reporting PDA) by defining missing values as NAs.

# Assign NAs strings to studies with missing event data  
ca\_d\_full\_3m <- ca\_d\_full\_3m%>%  
 mutate(across(c(e\_abs, e\_non\_abs, e\_n, c\_abs, c\_non\_abs, c\_n), ~ ifelse(is.na(.), "NA", as.character(.))))

To make it clearer in the final forest plot, we assign new variable names to the two subgroups, and sort them with the subgroup with the lowest number of studies on top.

# Assigning new variable names to the subgroups  
ca\_d\_full\_3m <- ca\_d\_full\_3m %>%  
 dplyr::rename(Subgroup = super\_domain) %>%  
 dplyr::mutate( # Changing the order of the two subgroups  
 Subgroup = factor(Subgroup,  
 levels = c("Explicit", "Implicit"),  
 labels = c("Explicit cognition", "Implicit cognition"))  
 )

With the new subgroup names, we can pool the effects across studies and stratify by the superordinate domains targeted by the cognitive training programs (i.e., explicit and implicit cognition) used in each study. This is achieved by using metagen(), where we define our treatment effect (TE) as the lnOR and corresponding standard error of the treatment effect (seTE) as the SE of the lnOR. We apply a random effects model using REML and use a classic *z*-test.

# Run the meta-analysis stratified by superordinate domains and using the new labels  
ca\_metagen\_3m <- metagen(TE = lnOR,  
 seTE = SElnOR,  
 studlab = study,  
 data = ca\_d\_full\_3m,  
 subgroup = Subgroup,  
 sm = "OR",  
 method.tau = "REML",  
 common = FALSE,  
 random = TRUE,  
 method.random.ci = "classic")  
  
# Print the results of the meta-analysis  
print(ca\_metagen\_3m)

## Number of studies: k = 11  
##   
## OR 95%-CI z p-value  
## Random effects model 1.2396 [1.0534; 1.4587] 2.59 0.0097  
##   
## Quantifying heterogeneity (with 95%-CIs):  
## tau^2 = 0.0047 [0.0000; 1.1595]; tau = 0.0687 [0.0000; 1.0768]  
## I^2 = 39.6% [0.0%; 70.3%]; H = 1.29 [1.00; 1.83]  
##   
## Test of heterogeneity:  
## Q d.f. p-value  
## 16.57 10 0.0845  
##   
## Results for subgroups (random effects model):  
## k OR 95%-CI tau^2 tau Q  
## Subgroup = Explicit cognition 1 7.1111 [1.9858; 25.4649] -- -- 0.00  
## Subgroup = Implicit cognition 10 1.2103 [1.0450; 1.4017] <0.0001 0.0012 9.26  
## I^2  
## Subgroup = Explicit cognition --  
## Subgroup = Implicit cognition 2.8%  
##   
## Test for subgroup differences (random effects model):  
## Q d.f. p-value  
## Between groups 7.31 1 0.0069  
##   
## Details of meta-analysis methods:  
## - Inverse variance method  
## - Restricted maximum-likelihood estimator for tau^2  
## - Q-Profile method for confidence interval of tau^2 and tau  
## - Calculation of I^2 based on Q

The test statistics for the overall pooled effect size as well as the subgroup effect sizes have now been calculated, but by default, these are not displayed in the forest plot created with the forest()-function in the meta-package. These values can be displayed, if we first extract the test statistics and store them as separate data-frames. The data frames can then be added manually to the forest plot with the grid-package.

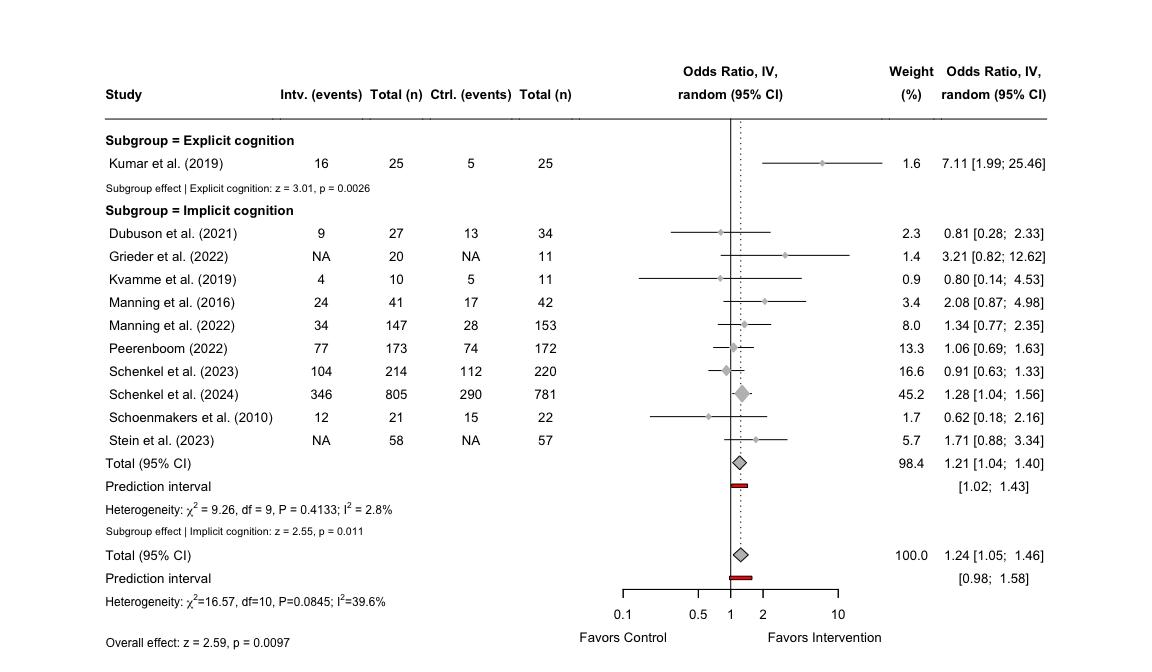
# Extract the test statistics for the overall effect and create a data frame  
ca\_zp\_overall\_3m <- data.frame(  
 lnOR = ca\_metagen\_3m$TE.random,  
 lnOR\_se = ca\_metagen\_3m$seTE.random,  
 z\_value = ca\_metagen\_3m$statistic.random,  
 p\_value = ca\_metagen\_3m$pval.random  
)  
  
# Extract the test statistics for the subgroup analysis and create a data frame  
ca\_zp\_subgroup\_3m <- data.frame(  
 subgroup = ca\_metagen\_3m$bylevs,  
 lnOR = ca\_metagen\_3m$TE.random.w,  
 lnOR\_se = ca\_metagen\_3m$seTE.random.w,  
 z\_value = ca\_metagen\_3m$statistic.random.w,  
 p\_value = ca\_metagen\_3m$pval.random.w  
)

For an easier way to add the individual effect sizes for each subgroup in the final forest plot, we can create a split of the data frame.

# Split the previously defined data frame by each superordinate domain  
ca\_zp\_subsplit\_3m <- split(ca\_zp\_subgroup\_3m, ca\_zp\_subgroup\_3m$subgroup)

With the final preparation completed, we can plot the forest plot using the random effects model with the additional test statistics added with the grid-package. The forest plot uses the BMJ layout (i.e., added weights for the individual studies) and it shows the number of events (i.e., abstinents), the group size for both the intervention (i.e., Intv.) and the control group (i.e., Ctrl), OR effects for individual studies across subgroups as well as the overall effect. A prediction interval is added to the pooled effect sizes. For the heterogeneity analyses, we report the Cochran’s *Q* statistic, its *p*-value as well as the measure of absolute variance using .

# Now plot the forest plot again with new subgroup names  
forest(ca\_metagen\_3m,  
 prediction = TRUE,  
 prediction.subgroup = TRUE,  
 layout = "BMJ",  
 subgroup = TRUE,  
 overall = TRUE,  
 random = TRUE,  
 common = FALSE,  
 test.subgroup = FALSE,  
 print.stat = FALSE,  
 print.I2 = TRUE,  
 print.Q = TRUE,  
 print.pval.Q = TRUE,  
 print.tau2 = FALSE,  
 label.left = "Favors Control",  
 label.right = "Favors Intervention",  
 spacing = 1.2,  
 fontsize = 10,  
 digits = 2,  
 leftcols = c("studlab", "e\_abs", "e\_n", "c\_abs", "c\_n"),  
 leftlabs = c("Study", "Intv. (events)", "Total (n)",   
 "Ctrl. (events)", "Total (n)"),  
 just.addcols = "center"  
 )  
  
  
# Add a footer for the test statistics of the effects for explicit cognition  
footer\_exp\_ca\_3m <- paste0(  
 "Subgroup effect | ",  
 paste0(  
 ca\_zp\_subsplit\_3m$`Explicit cognition`,  
 ": z = ", round(ca\_zp\_subsplit\_3m$`Explicit cognition`$z\_value, 2),  
 ", p = ", formatC(ca\_zp\_subsplit\_3m$`Explicit cognition`$p\_value, digits = 2)  
 )  
)  
  
# Add a footer for the test statistics of the effects for implicit cognition  
footer\_imp\_ca\_3m <- paste0(  
 "Subgroup effect | ",  
 paste0(  
 ca\_zp\_subsplit\_3m$`Implicit cognition`,  
 ": z = ", round(ca\_zp\_subsplit\_3m$`Implicit cognition`$z\_value, 2),  
 ", p = ", formatC(ca\_zp\_subsplit\_3m$`Implicit cognition`$p\_value, digits = 2)  
 )  
)  
  
# Add a footer combining the test statistics for the overall effect  
footer\_over\_ca\_3m <- paste0(  
 "Overall effect: ",  
 paste0(  
 "z = ", round(ca\_zp\_overall\_3m$z\_value, 2),  
 ", p = ", formatC(ca\_zp\_overall\_3m$p\_value, digits = 2),  
 collapse = "; "  
 )  
)  
  
# Adjust the footer for the subgroup effects after explicit subgroup  
grid.text(footer\_exp\_ca\_3m, x = 0.092, y = 0.72,  
 just = "left", gp = gpar(fontsize = 8))  
  
# Adjust the footer for the subgroup effects after implicit subgroup  
grid.text(footer\_imp\_ca\_3m, x = 0.092, y = 0.21,  
 just = "left", gp = gpar(fontsize = 8))  
  
# Adjust the footer for the overall effect to the bottom left  
grid.text(footer\_over\_ca\_3m, x = 0.092, y = 0.045,  
 just = "left", gp = gpar(fontsize = 9))



1. Forest plot for the effects on continuous abstinence

### 2.1.5 Sensitivity analysis for continuous abstinence

In the primary analysis, we included two studies that did not report a categorical outcome for continuous abstinence, which can introduce measurement error and increase the heterogeneity. Thus, we also conduct a sensitivity analysis, where these two studies are excluded, to see how this affects the overall results. To achieve this, we can start by filtering out these two studies by the study label in our previously defined data frame.

# Exclude the study by Grieder and Stein and store a new data frame  
ca\_d\_sens\_filter\_3m <- ca\_d\_full\_3m %>%   
 filter(!study %in% c("Grieder et al. (2022)", "Stein et al. (2023)"))

We want to re-arrange the study labels in alphabetical order rather then by the pre-defined row numbers.

# Arrange studies alphabetically  
ca\_d\_sens\_filter\_3m <- ca\_d\_sens\_filter\_3m %>%  
 arrange(desc(study == "Dubuson et al. (2021)"),  
 desc(study == "Kvamme et al. (2019)"),  
 desc(study == "Manning et al. (2016)"),  
 desc(study == "Manning et al. (2022)"),  
 desc(study == "Peerenboom (2022)"),  
 desc(study == "Schenkel et al. (2023)"),  
 desc(study == "Schenkel et al. (2024)"),  
 desc(study == "Schoenmakers et al. (2010)")  
 )

Now that the data set has been filtered, we can run the metagen()-function again, this time with our filtered data frame. We still want to run the analysis stratified by the previously defined subgroups.

# Run the meta-analysis stratified by superordinate domains  
ca\_sens\_metagen\_3m <- metagen(TE = lnOR,  
 seTE = SElnOR,  
 studlab = study,  
 data = ca\_d\_sens\_filter\_3m,  
 subgroup = Subgroup,  
 sm = "OR",  
 method.tau = "REML",  
 common = FALSE,  
 random = TRUE,  
 method.random.ci = "classic")  
  
# Print the results of the meta-analysis  
print(ca\_sens\_metagen\_3m)

## Number of studies: k = 9  
##   
## OR 95%-CI z p-value  
## Random effects model 1.1970 [1.0072; 1.4225] 2.04 0.0412  
##   
## Quantifying heterogeneity (with 95%-CIs):  
## tau^2 = 0.0058 [0.0000; 1.5640]; tau = 0.0764 [0.0000; 1.2506]  
## I^2 = 41.5% [0.0%; 73.0%]; H = 1.31 [1.00; 1.93]  
##   
## Test of heterogeneity:  
## Q d.f. p-value  
## 13.67 8 0.0907  
##   
## Results for subgroups (random effects model):  
## k OR 95%-CI tau^2 tau Q  
## Subgroup = Explicit cognition 1 7.1111 [1.9858; 25.4649] -- -- 0.00  
## Subgroup = Implicit cognition 8 1.1616 [0.9808; 1.3757] 0.0044 0.0661 6.12  
## I^2  
## Subgroup = Explicit cognition --  
## Subgroup = Implicit cognition 0.0%  
##   
## Test for subgroup differences (random effects model):  
## Q d.f. p-value  
## Between groups 7.62 1 0.0058  
##   
## Details of meta-analysis methods:  
## - Inverse variance method  
## - Restricted maximum-likelihood estimator for tau^2  
## - Q-Profile method for confidence interval of tau^2 and tau  
## - Calculation of I^2 based on Q

The test statistics for the newly calculated effect sizes can now be extracted, and prepared for our forest plot.

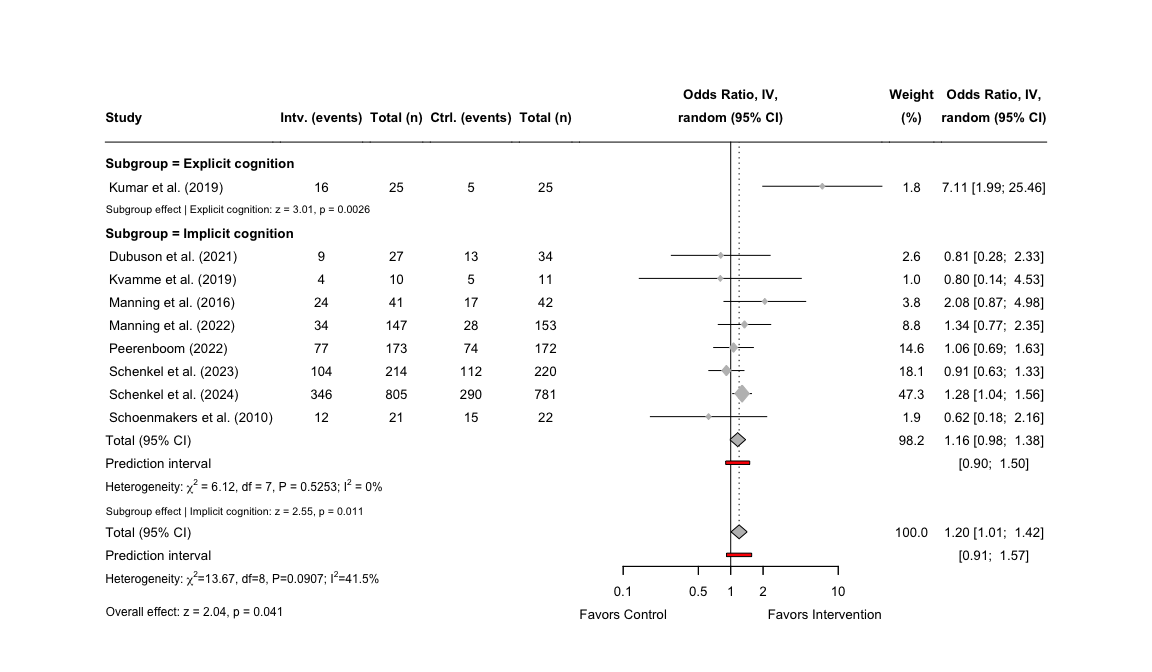
# Extract the test statistics for the overall effect and create a data frame  
ca\_zp\_sens\_over\_3m <- data.frame(  
 lnOR = ca\_sens\_metagen\_3m$TE.random,  
 lnOR\_se = ca\_sens\_metagen\_3m$seTE.random,  
 z\_value = ca\_sens\_metagen\_3m$statistic.random,  
 p\_value = ca\_sens\_metagen\_3m$pval.random  
)  
  
# Extract the test statistics for the subgroup analysis and create a data frame  
ca\_zp\_sens\_sub\_3m <- data.frame(  
 subgroup = ca\_sens\_metagen\_3m$bylevs,  
 lnOR = ca\_sens\_metagen\_3m$TE.random.w,  
 lnOR\_se = ca\_sens\_metagen\_3m$seTE.random.w,  
 z\_value = ca\_sens\_metagen\_3m$statistic.random.w,  
 p\_value = ca\_sens\_metagen\_3m$pval.random.w  
)

For an easier way to add the individual effect sizes for each subgroup in the final forest plot, we can create a split of the data frame.

# Split the previously defined data frame by each superordinate domain  
ca\_zp\_sens\_subsplit\_3m <- split(ca\_zp\_sens\_sub\_3m, ca\_zp\_sens\_sub\_3m$subgroup)

Now with the extracted test statistics, we can now plot the results of the sensitivity analysis and add additional information using the grid-package. We use the same layout and measures for heterogeneity, which was used for our unadjusted analysis.

# Forest plot for sensitivity analysis  
forest(ca\_sens\_metagen\_3m,  
 prediction = TRUE,  
 prediction.subgroup = TRUE,  
 layout = "BMJ",  
 subgroup = TRUE,  
 overall = TRUE,  
 random = TRUE,  
 common = FALSE,  
 test.subgroup = FALSE,  
 print.stat = FALSE,  
 print.I2 = TRUE,  
 print.Q = TRUE,  
 print.pval.Q = TRUE,  
 print.tau2 = FALSE,  
 label.left = "Favors Control",  
 label.right = "Favors Intervention",  
 spacing = 1.2,  
 fontsize = 10,  
 digits = 2,  
 leftcols = c("studlab", "e\_abs", "e\_n", "c\_abs", "c\_n"),  
 leftlabs = c("Study", "Intv. (events)", "Total (n)",   
 "Ctrl. (events)", "Total (n)"),  
 just.addcols = "center"  
 )  
  
  
# Add a footer for the test statistics of the effects for explicit cognition  
footer\_exp\_ca\_sens\_3m <- paste0(  
 "Subgroup effect | ",  
 paste0(  
 ca\_zp\_sens\_subsplit\_3m$`Explicit cognition`,  
 ": z = ", round(ca\_zp\_sens\_subsplit\_3m$`Explicit cognition`$z\_value, 2),  
 ", p = ", formatC(ca\_zp\_sens\_subsplit\_3m$`Explicit cognition`$p\_value, digits = 2)  
 )  
)  
  
# Add a footer for the test statistics of the effects for implicit cognition  
footer\_imp\_ca\_sens\_3m <- paste0(  
 "Subgroup effect | ",  
 paste0(  
 ca\_zp\_subsplit\_3m$`Implicit cognition`,  
 ": z = ", round(ca\_zp\_subsplit\_3m$`Implicit cognition`$z\_value, 2),  
 ", p = ", formatC(ca\_zp\_subsplit\_3m$`Implicit cognition`$p\_value, digits = 2)  
 )  
)  
  
# Add a footer combining the test statistics for the overall effect  
footer\_over\_ca\_sens\_3m <- paste0(  
 "Overall effect: ",  
 paste0(  
 "z = ", round(ca\_zp\_sens\_over\_3m$z\_value, 2),  
 ", p = ", formatC(ca\_zp\_sens\_over\_3m$p\_value, digits = 2),  
 collapse = "; "  
 )  
)  
  
# Adjust the footer for the subgroup effects after explicit subgroup  
grid.text(footer\_exp\_ca\_sens\_3m, x = 0.092, y = 0.69,  
 just = "left", gp = gpar(fontsize = 8))  
  
# Adjust the footer for the subgroup effects after implicit subgroup  
grid.text(footer\_imp\_ca\_sens\_3m, x = 0.092, y = 0.24,  
 just = "left", gp = gpar(fontsize = 8))  
  
# Adjust the footer for the overall effect to the bottom left  
grid.text(footer\_over\_ca\_sens\_3m, x = 0.092, y = 0.09,  
 just = "left", gp = gpar(fontsize = 9))



1. Forest plot for the sensitivity analysis on continuous abstinence

### 2.1.6 Publication bias for short-term effects on continuous abstinence

The final step for the primary analyses on continuous abstinence, is to assess the publication bias of the included studies. We do this by conducting a meta-analysis once more, but this time we use the metagen()-function without stratifying for the subgroups. We use the previously stored data frame combining categorical and continuous outcomes, and set the treatment effect to lnOR and the standard error of the treatment effect to SElnOR.

# Run meta-analysis without stratifying for subgroups  
ca\_pupbias\_3m <- metagen(TE = lnOR,  
 seTE = SElnOR,  
 studlab = study,  
 sm = "OR",  
 method.tau = "REML",  
 data = ca\_d\_full\_3m)  
  
# Print the results of the meta-analysis without subgroups  
print(ca\_pupbias\_3m)

## Number of studies: k = 11  
##   
## OR 95%-CI z p-value  
## Common effect model 1.2386 [1.0705; 1.4331] 2.88 0.0040  
## Random effects model 1.2396 [1.0534; 1.4587] 2.59 0.0097  
##   
## Quantifying heterogeneity (with 95%-CIs):  
## tau^2 = 0.0047 [0.0000; 1.1595]; tau = 0.0687 [0.0000; 1.0768]  
## I^2 = 39.6% [0.0%; 70.3%]; H = 1.29 [1.00; 1.83]  
##   
## Test of heterogeneity:  
## Q d.f. p-value  
## 16.57 10 0.0845  
##   
## Details of meta-analysis methods:  
## - Inverse variance method  
## - Restricted maximum-likelihood estimator for tau^2  
## - Q-Profile method for confidence interval of tau^2 and tau  
## - Calculation of I^2 based on Q

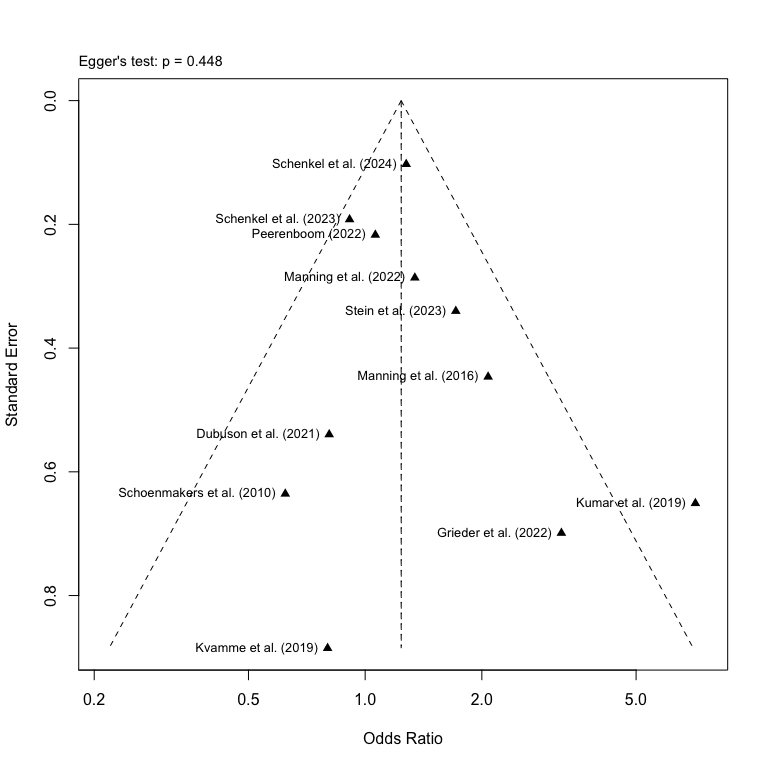
As we want to both be able to visually inspect our plot for asymmetry as well as having a method of quantifying publication bias, we first want to conduct an Egger’s test, which can be added to the final funnel plot.

# Conduct an Egger's test using the previosly calculated treatment effects  
ca\_egger\_3m <- metabias(ca\_pupbias\_3m, method.bias = "Egger")  
  
# Print the test results of the Egger's test  
print(ca\_egger\_3m)

## Linear regression test of funnel plot asymmetry  
##   
## Test result: t = 0.79, df = 9, p-value = 0.4483  
## Bias estimate: 0.5250 (SE = 0.6623)  
##   
## Details:  
## - multiplicative residual heterogeneity variance (tau^2 = 1.7209)  
## - predictor: standard error  
## - weight: inverse variance  
## - reference: Egger et al. (1997), BMJ

Now with the results from the Egger’s test, we can now create the funnel plot using the meta-package inbuilt function funnel() and exponentiate the lnOR values to have the x-axis displaying ORs instead. We can add the test statistic from the Egger’s test to the funnel plot using mtext().

# Create a funnel plot using the newly stored data frame  
funnel(ca\_pupbias\_3m,  
 xlab = "Odds Ratio",  
 studlab = TRUE,  
 backtransf = TRUE,  
 pch = 17,  
 cex = 1)  
  
# Add Egger's test statistic to the plot  
ca\_egger\_text\_3m <- paste0("Egger's test: p = ",   
 round(ca\_egger\_3m$p.value, 3))  
  
# Define the actual placement of the statistics on the funnel plot  
mtext(ca\_egger\_text\_3m, side=3, line=0.5, adj=0, cex=0.9, col="black")



1. Funnel plot for studies on short-term effects on continuous abstinence

## 2.2 Short-term effects on alcohol reduction

The first part of the primary analysis was on our primary outcome CA, now we can turn to analysing our secondary outcome on alcohol reduction. We still need to use the excel-dataset “Extracted Clinical Outcome Data”, but this time we import the sheet named “Reduction” and assign row numbers for an easier sorting of the studies.

# Import data on abstinence as "ca\_d" and preserve original order of studies  
red\_d <- read\_excel(osf\_clinical\_files$local\_path, sheet = "Reduction") %>%  
 mutate(row\_id = row\_number())

### 2.2.1 Calculation of effect sizes for continuous outcomes

For the primary analysis on alcohol reduction, we still want to focus on the time point “0-3 months”, so we filter the data set by time point and re-arrange the studies in terms of alphabetical order.

# Filter 0–3 month data and arrange specific studies first  
red\_d\_3m <- red\_d %>%  
 filter(time\_point == "0-3 months") %>%  
 arrange(desc(study == "7c\_garfield\_2024"),  
 desc(study == "10b\_laurens\_2023"),  
 desc(study == "11\_manning\_2016"))

As we only have continuous data for alcohol reduction, we only need to run the metacont() on our dataset to calculate the SMDs using Hedge’s *g* correction.

# Calculating the SMDs (Hedge's g) using metacont()  
red\_metacont\_3m <- metacont(  
 n.e = e\_n,  
 mean.e = e\_reduc\_mean,  
 sd.e = e\_reduc\_sd,  
 n.c = c\_n,  
 mean.c = c\_reduc\_mean,  
 sd.c = c\_reduc\_sd,  
 studlab = study,  
 data = red\_d\_3m,  
 sm = "SMD",  
 method.smd = "Hedges",  
 method.tau = "REML",  
 method.random.ci = "classic",  
 random = TRUE  
)

Now we can combine the calculated effect sizes and add them to the original data set without changing the original variable names.

# Combine the SMDs calculated with original variables and store it in a new data frame  
red\_d\_full\_3m <- data.frame(  
 row\_id = red\_d\_3m$row\_id,  
 study = red\_metacont\_3m$studlab,  
 time\_point = red\_d\_3m$time\_point,  
 super\_domain = red\_d\_3m$super\_domain,  
 cog\_domain = red\_d\_3m$cog\_domain,  
 total\_n = red\_d\_3m$e\_n + red\_d\_3m$c\_n, # Calculating total number of participants  
 c\_n = red\_d\_3m$c\_n,  
 c\_reduc\_mean = red\_d\_3m$c\_reduc\_mean,  
 c\_reduc\_sd = red\_d\_3m$c\_reduc\_sd,  
 e\_n = red\_d\_3m$e\_n,  
 e\_reduc\_mean = red\_d\_3m$e\_reduc\_mean,  
 e\_reduc\_sd = red\_d\_3m$e\_reduc\_sd,  
 # Adding calculated SMDs to the dataframe   
 SMD = red\_metacont\_3m$TE,  
 SE = red\_metacont\_3m$seTE,  
 LCI = red\_metacont\_3m$lower,  
 UCI = red\_metacont\_3m$upper,  
 # Adding the z- and p-values  
 `z-value` = red\_metacont\_3m$TE / red\_metacont\_3m$seTE,  
 `p-value` = 2 \* (1 - pnorm(abs(red\_metacont\_3m$TE / red\_metacont\_3m$seTE))),  
 # Adding a final note on how data was converted   
 Notes = "Calculated using metacont()",  
 check.names = FALSE # Removing the period as a space  
)

### 2.2.2 Meta-analysis on alcohol reduction

After we have calculated the effect sizes and added them to the original data set, we can now assign the previously defined study labels to the data set.

# Assign the new study labels to the combined data frame  
red\_d\_full\_3m$study <- ifelse(red\_d\_full\_3m$study %in% names(study\_labels),  
 study\_labels[red\_d\_full\_3m$study],  
 red\_d\_full\_3m$study)

It is now possible to run the meta-analysis for alcohol reduction using the metagen() function, setting the Hedges *g* correction for the SMD and REML for the heterogeneity analysis. However, as no study examined the effects of explicit cognitive training on alcohol reduction, we do not apply stratification by subgroups.

# Run the meta-analysis stratified by superordinate domains and using the new labels  
red\_metagen\_3m <- metagen(TE = SMD,  
 seTE = SE,  
 studlab = study,  
 data = red\_d\_full\_3m,  
 sm = "SMD",  
 method.smd = "Hedges",  
 method.tau = "REML",  
 common = FALSE,  
 random = TRUE,  
 method.random.ci = "classic")  
  
# Print the results of the meta-analysis  
print(red\_metagen\_3m)

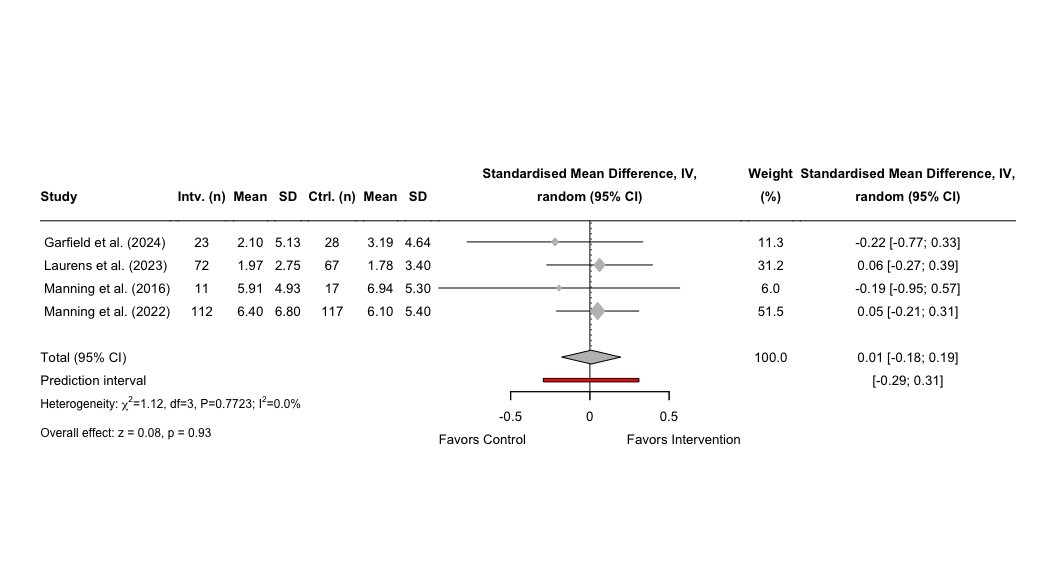
## Number of studies: k = 4  
##   
## SMD 95%-CI z p-value  
## Random effects model 0.0078 [-0.1782; 0.1938] 0.08 0.9345  
##   
## Quantifying heterogeneity (with 95%-CIs):  
## tau^2 = 0 [0.0000; 0.2560]; tau = 0 [0.0000; 0.5059]  
## I^2 = 0.0% [0.0%; 84.7%]; H = 1.00 [1.00; 2.56]  
##   
## Test of heterogeneity:  
## Q d.f. p-value  
## 1.12 3 0.7723  
##   
## Details of meta-analysis methods:  
## - Inverse variance method  
## - Restricted maximum-likelihood estimator for tau^2  
## - Q-Profile method for confidence interval of tau^2 and tau  
## - Calculation of I^2 based on Q

The test statistics can now be extracted from the meta-analysis and stored as a data frame for the forest plot.

# Extract the test statistics for the overall effect and create a data frame  
red\_zp\_over\_3m <- data.frame(  
 SMD = red\_metagen\_3m$TE.random,  
 SE = red\_metagen\_3m$seTE.random,  
 z\_value = red\_metagen\_3m$statistic.random,  
 p\_value = red\_metagen\_3m$pval.random  
)

With the test statistics stored as a data frame, we now turn to creating the forest plot, which follows the same design as previously used for CA, however, no subgroup labels will be present, as only effects of implicit cognitive training on reduction was examined by the identified studies.

# Now plot the forest plot again with new subgroup names  
forest(red\_metagen\_3m,  
 prediction = TRUE,  
 layout = "BMJ",  
 overall = TRUE,  
 random = TRUE,  
 common = FALSE,  
 print.stat = FALSE,  
 print.I2 = TRUE,  
 print.Q = TRUE,  
 print.pval.Q = TRUE,  
 print.tau2 = FALSE,  
 label.left = "Favors Control",  
 label.right = "Favors Intervention",  
 spacing = 1.2,  
 fontsize = 10,  
 digits = 2,  
 leftcols = c("studlab", "e\_n", "e\_reduc\_mean", "e\_reduc\_sd", "c\_n", "c\_reduc\_mean",  
 "c\_reduc\_sd"),  
 leftlabs = c("Study", "Intv. (n)", "Mean", "SD",   
 "Ctrl. (n)", "Mean", "SD"),  
 just.addcols = "center"  
 )  
  
# Add a footer combining the test statistics for the overall effect  
footer\_over\_red\_3m <- paste0(  
 "Overall effect: ",  
 paste0(  
 "z = ", round(red\_zp\_over\_3m$z\_value, 2),  
 ", p = ", formatC(red\_zp\_over\_3m$p\_value, digits = 2),  
 collapse = "; "  
 )  
)  
  
# Adjust the footer for the overall effect to the bottom left  
grid.text(footer\_over\_red\_3m, x = 0.0385, y = 0.25,  
 just = "left", gp = gpar(fontsize = 9))



1. Forest plot of effects on alcohol reduction

### 2.2.3 Publication bias for short-term effects on alcohol reduction

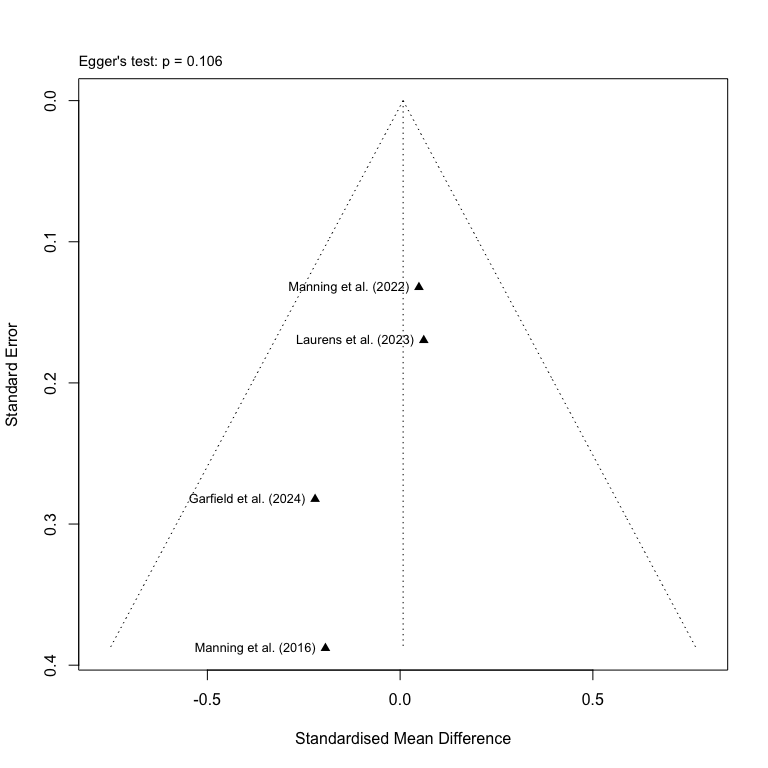
The final step for the primary analyses on alcohol reduction, is to assess the publication bias of the included studies. We can reuse the meta-analyses conducted in the section above for the funnel plot. Instead, we can run a Egger’s test, but due to a low number of studies, we need to set the minimum *k* to three studies. The results will, however, be less reliable given the small number of studies, and interpretation of the results should be cautioned.

# Conduct an Egger's test using the previously calculated treatment effects  
red\_egger\_3m <- metabias(red\_metagen\_3m, method.bias = "Egger", k.min = 3)  
  
# Print the results of the Egger's test  
print(red\_egger\_3m)

## Linear regression test of funnel plot asymmetry  
##   
## Test result: t = -2.83, df = 2, p-value = 0.1056  
## Bias estimate: -1.2746 (SE = 0.4508)  
##   
## Details:  
## - multiplicative residual heterogeneity variance (tau^2 = 0.1120)  
## - predictor: standard error  
## - weight: inverse variance  
## - reference: Egger et al. (1997), BMJ

Now with the results from the Egger’s test, we can now create the funnel plot using the meta-package inbuilt function funnel(). We can add the test statistic from the Egger’s test to the funnel plot using mtext().

# Create a funnel plot using the newly stored data frame  
funnel(red\_metagen\_3m,  
 xlab = "Standardised Mean Difference",  
 studlab = TRUE,  
 pch = 17,  
 cex = 1)  
  
# Add Egger's test statistic to the plot  
red\_egger\_text\_3m <- paste0("Egger's test: p = ",   
 round(red\_egger\_3m$p.value, 3))  
  
# Define the actual placement of the statistics on the funnel plot  
mtext(red\_egger\_text\_3m, side=3, line=0.5, adj=0, cex=0.9, col="black")



1. Funnel plot on studies for short-term effects on alcohol reduction

## 2.3 Short-term effects on alcohol craving

The final part of the primary analysis will be conducted on alcohol craving. We still need to use the excel-dataset “Extracted Clinical Outcome Data”, but this time we import the sheet named “Craving” and assign row numbers for an easier sorting of the studies.

# Import data on reduction as "crav\_d" and preserve original order of studies  
crav\_d <- read\_excel(osf\_clinical\_files$local\_path, sheet = "Craving") %>%   
 mutate(row\_id = row\_number())

### 2.3.1 Calculation of effect sizes for continuous outcomes

For the primary analysis on alcohol reduction, we still want to focus on the time point “0-3 months”, so we filter the data set by time point and re-arrange the studies in terms of alphabetical order.

# Filter 0–3 month data and arrange specific studies first  
crav\_d\_3m <- crav\_d %>%  
 filter(time\_point == "0-3 months") %>%  
 arrange(desc(study == "3a\_dubuson\_2021"),  
 desc(study == "5f\_garfield\_2022"),  
 desc(study == "8b\_kvamme\_2019"),  
 desc(study == "10b\_laurens\_2023"),  
 desc(study == "11\_manning\_2016"),  
 desc(study == "16a\_schoenmakers\_2010"),  
 desc(study == "17b\_stein\_2023"),  
 desc(study == "2a\_uyl\_2018"),  
 desc(study == "18a\_wiers\_2011")  
 )

As we only have continuous data for alcohol craving, we only need to run the metacont()-function on our dataset to calculate the SMDs using Hedge’s *g* correction.

# Calculating the SMDs (Hedge's g) using metacont()  
crav\_metacont\_3m <- metacont(  
 n.e = e\_n,  
 mean.e = e\_crav\_mean,  
 sd.e = e\_crav\_sd,  
 n.c = c\_n,  
 mean.c = c\_crav\_mean,  
 sd.c = c\_crav\_sd,  
 studlab = study,  
 data = crav\_d\_3m,  
 sm = "SMD",  
 method.smd = "Hedges",  
 method.tau = "REML",  
 method.random.ci = "classic",  
 random = TRUE  
)

Now we can combine the calculated effect sizes and add them to the original data set without changing the original variable names.

# Combine the SMDs calculated with original variables and store it in a new data frame  
crav\_d\_full\_3m <- data.frame(  
 row\_id = crav\_d\_3m$row\_id,  
 study = crav\_metacont\_3m$studlab,  
 time\_point = crav\_d\_3m$time\_point,  
 super\_domain = crav\_d\_3m$super\_domain,  
 cog\_domain = crav\_d\_3m$cog\_domain,  
 total\_n = crav\_d\_3m$e\_n + crav\_d\_3m$c\_n, # Calculating total number of participants  
 c\_n = crav\_d\_3m$c\_n,  
 c\_crav\_mean = crav\_d\_3m$c\_crav\_mean,  
 c\_crav\_sd = crav\_d\_3m$c\_crav\_sd,  
 e\_n = crav\_d\_3m$e\_n,  
 e\_crav\_mean = crav\_d\_3m$e\_crav\_mean,  
 e\_crav\_sd = crav\_d\_3m$e\_crav\_sd,  
 # Adding calculated SMDs to the dataframe   
 SMD = crav\_metacont\_3m$TE,  
 SE = crav\_metacont\_3m$seTE,  
 LCI = crav\_metacont\_3m$lower,  
 UCI = crav\_metacont\_3m$upper,  
 # Adding the z- and p-values  
 `z-value` = crav\_metacont\_3m$TE / crav\_metacont\_3m$seTE,  
 `p-value` = 2 \* (1 - pnorm(abs(crav\_metacont\_3m$TE / crav\_metacont\_3m$seTE))),  
 # Adding a final note on how data was converted   
 Notes = "Calculated using metacont()",  
 check.names = FALSE # Removing the period as a space  
)

### 2.3.2 Meta-analysis on alcohol craving

After we have calculated the effect sizes and added them to the original data set, we can now assign the previously defined study labels to the data set.

# Assign the new study labels to the combined data frame  
crav\_d\_full\_3m$study <- ifelse(crav\_d\_full\_3m$study %in% names(study\_labels),  
 study\_labels[crav\_d\_full\_3m$study],  
 crav\_d\_full\_3m$study)

It is now possible to run the meta-analysis for alcohol craving using the metagen() function, setting the Hedges *g* correction for the SMD and REML for the heterogeneity analysisl. However, as no study examined the effects of explicit cognitive training on alcohol craving, we do not apply stratification by subgroups.

# Run the meta-analysis stratified by superordinate domains and using the new labels  
crav\_metagen\_3m <- metagen(TE = SMD,  
 seTE = SE,  
 studlab = study,  
 data = crav\_d\_full\_3m,  
 sm = "SMD",  
 method.smd = "Hedges",  
 method.tau = "REML",  
 common = FALSE,  
 random = TRUE,  
 method.random.ci = "classic")  
  
# Print the results of the meta-analysis  
print(crav\_metagen\_3m)

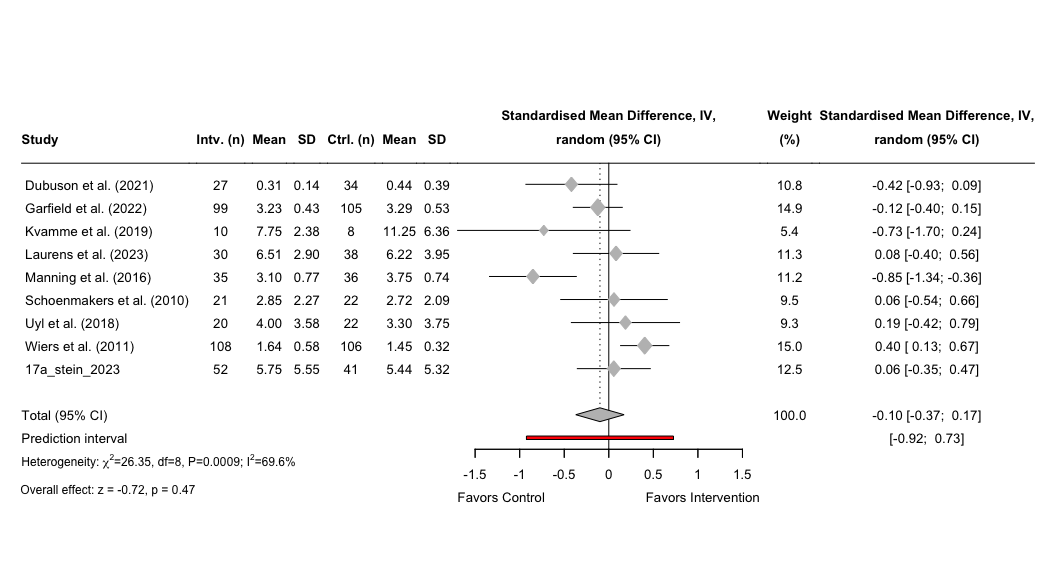
## Number of studies: k = 9  
##   
## SMD 95%-CI z p-value  
## Random effects model -0.0992 [-0.3703; 0.1720] -0.72 0.4734  
##   
## Quantifying heterogeneity (with 95%-CIs):  
## tau^2 = 0.1089 [0.0200; 0.5794]; tau = 0.3300 [0.1414; 0.7612]  
## I^2 = 69.6% [39.5%; 84.8%]; H = 1.81 [1.29; 2.56]  
##   
## Test of heterogeneity:  
## Q d.f. p-value  
## 26.35 8 0.0009  
##   
## Details of meta-analysis methods:  
## - Inverse variance method  
## - Restricted maximum-likelihood estimator for tau^2  
## - Q-Profile method for confidence interval of tau^2 and tau  
## - Calculation of I^2 based on Q

The test statistics can now be extracted from the meta-analysis and stored as a data frame for the forest plot.

# Extract the test statistics for the overall effect and create a data frame  
crav\_zp\_over\_3m <- data.frame(  
 SMD = crav\_metagen\_3m$TE.random,  
 SE = crav\_metagen\_3m$seTE.random,  
 z\_value = crav\_metagen\_3m$statistic.random,  
 p\_value = crav\_metagen\_3m$pval.random  
)

With the test statistics stored as a data frame, we now turn to creating the forest plot, which follows the same design as previously used, however, no subgroup labels will be present, as only effects of implicit cognitive training on reduction was examined by the identified studies.

# Now plot the forest plot again with new subgroup names  
forest(crav\_metagen\_3m,  
 prediction = TRUE,  
 layout = "BMJ",  
 overall = TRUE,  
 random = TRUE,  
 common = FALSE,  
 print.stat = FALSE,  
 print.I2 = TRUE,  
 print.Q = TRUE,  
 print.pval.Q = TRUE,  
 print.tau2 = FALSE,  
 label.left = "Favors Control",  
 label.right = "Favors Intervention",  
 spacing = 1.2,  
 fontsize = 10,  
 digits = 2,  
 leftcols = c("studlab", "e\_n", "e\_crav\_mean", "e\_crav\_sd", "c\_n", "c\_crav\_mean",  
 "c\_crav\_sd"),  
 leftlabs = c("Study", "Intv. (n)", "Mean", "SD",   
 "Ctrl. (n)", "Mean", "SD"),  
 just.addcols = "center"  
 )  
  
# Add a footer combining the test statistics for the overall effect  
footer\_over\_crav\_3m <- paste0(  
 "Overall effect: ",  
 paste0(  
 "z = ", round(crav\_zp\_over\_3m$z\_value, 2),  
 ", p = ", formatC(crav\_zp\_over\_3m$p\_value, digits = 2),  
 collapse = "; "  
 )  
)  
  
# Adjust the footer for the overall effect to the bottom left  
grid.text(footer\_over\_crav\_3m, x = 0.0195, y = 0.15,  
 just = "left", gp = gpar(fontsize = 9))



1. Forest plot of the effects on craving

### 2.3.3 Publication bias for short-term effects on craving

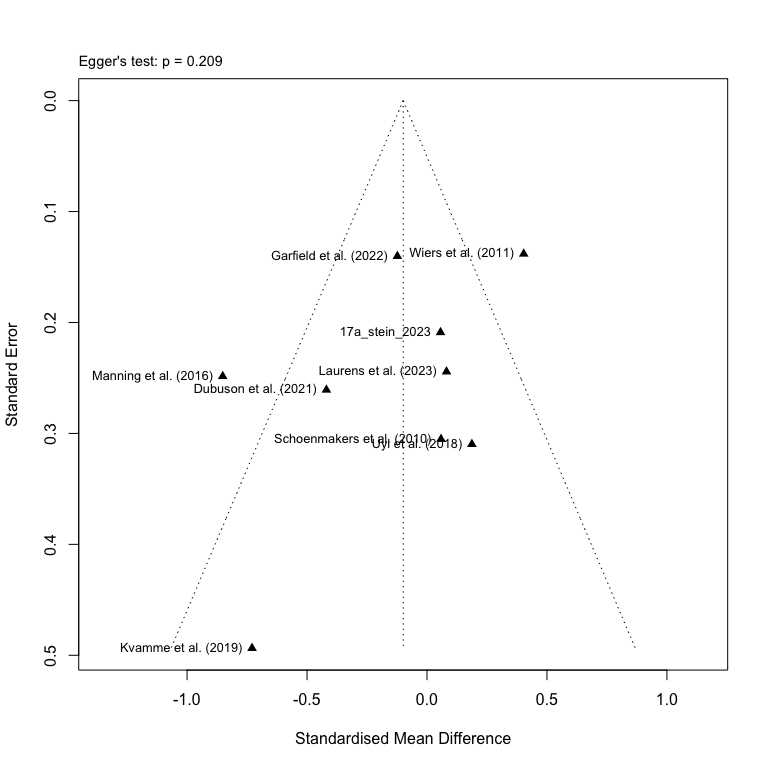
The final step for the primary analyses on alcohol craving, is to assess the publication bias of the included studies. We can reuse the meta-analyses conducted in the section above for the funnel plot. Instead, we need to run a Egger’s test, but due to a low number of studies, we need to set the minimum *k* to eight studies. The results will, however, be less reliable given the small number of studies, and interpretation of the results should be cautioned.

# Conduct an Egger's test using the previously calculated treatment effects  
crav\_egger\_3m <- metabias(crav\_metagen\_3m, method.bias = "Egger", k.min = 8)  
  
# Print the results of the Egger's test  
print(crav\_egger\_3m)

## Linear regression test of funnel plot asymmetry  
##   
## Test result: t = -1.38, df = 7, p-value = 0.2087  
## Bias estimate: -2.2617 (SE = 1.6336)  
##   
## Details:  
## - multiplicative residual heterogeneity variance (tau^2 = 2.9550)  
## - predictor: standard error  
## - weight: inverse variance  
## - reference: Egger et al. (1997), BMJ

Now with the results from the Egger’s test, we can now create the funnel plot using the meta-package inbuilt function funnel(). We can add the test statistic from the Egger’s test to the funnel plot using mtext(). As there is a high variability among these studies, we will need to adjust the limits of the x-axis so the studies can be better displayed.

# Calculate limits for x-axis to center the funnel  
smd\_summary <- crav\_metagen\_3m$TE.random # or TE.fixed if preferred  
xlim\_range <- max(abs(crav\_metagen\_3m$TE - smd\_summary)) + 0.5  
xlim <- c(smd\_summary - xlim\_range, smd\_summary + xlim\_range)  
  
# Create a funnel plot with adjusted x-axis  
funnel(crav\_metagen\_3m,  
 xlab = "Standardised Mean Difference",  
 studlab = TRUE,  
 pch = 17,  
 cex = 1,  
 xlim = xlim)  
  
# Add Egger's test statistic to the plot  
crav\_egger\_text\_3m <- paste0("Egger's test: p = ",   
 round(crav\_egger\_3m$p.value, 3))  
  
# Define the actual placement of the statistics on the funnel plot  
mtext(crav\_egger\_text\_3m, side=3, line=0.5, adj=0, cex=0.9, col="black")



1. Funnel plot of the studies on short-term effects on craving

# 3. Secondary analyses on short-term effects for subordinate domains (0-3 months)

The secondary analyses on short-term effects of cognitive training, will follow the same procedure and pool the effects for each of the specified outcomes, but this time they will be stratified by the subordinate cognitive domains targeted by each training program. We will still use the previously defined data frames for each of the outcomes and conduct the analyses with the new stratification.

## 3.1 Short-term effects on continuous abstinence by subgroups

To explore the short-term effects on CA by each subordinate domain, we first need to assign new variable names to the subgroups to create a visually more appealing forest plot. We assign labels to four subordinate domains: Executive functions (Explicit cognition), Approach bias (Implicit cognition), Attentional bias (Implicit cognition), and Inhibitory bias (Implicit cognition).

# Assigning new variable names to the subgroups  
ca\_d\_full\_3m <- ca\_d\_full\_3m %>%  
 dplyr::rename(super\_domain = Subgroup) %>%  
 dplyr::rename(Subgroup = cog\_domain) %>%  
 dplyr::mutate(  
 Subgroup = factor(  
 Subgroup,  
 levels = c("Executive functions", "Approach bias", "Attentional bias", "Inhibitory bias")  
 )  
 )

### 3.1.1 Meta-analysis on continuous abstinence

Now with the newly defined subgroup labels, we can conduct the meta-analysis using metagen(), but this time set the parameter for subgroups to the label “Subgroup” and use the classic *z*-test for the random effects model.

# Run the meta-analysis stratified by superordinate domains and using the new labels  
ca\_sec\_metagen\_3m <- metagen(TE = lnOR,  
 seTE = SElnOR,  
 studlab = study,  
 data = ca\_d\_full\_3m,  
 subgroup = Subgroup,  
 sm = "OR",  
 method.tau = "REML",  
 common = FALSE,  
 random = TRUE,  
 method.random.ci = "classic")  
  
# Print the results of the meta-analysis  
print(ca\_sec\_metagen\_3m)

## Number of studies: k = 11  
##   
## OR 95%-CI z p-value  
## Random effects model 1.2396 [1.0534; 1.4587] 2.59 0.0097  
##   
## Quantifying heterogeneity (with 95%-CIs):  
## tau^2 = 0.0047 [0.0000; 1.1595]; tau = 0.0687 [0.0000; 1.0768]  
## I^2 = 39.6% [0.0%; 70.3%]; H = 1.29 [1.00; 1.83]  
##   
## Test of heterogeneity:  
## Q d.f. p-value  
## 16.57 10 0.0845  
##   
## Results for subgroups (random effects model):  
## k OR 95%-CI tau^2 tau Q  
## Subgroup = Executive functions 1 7.1111 [1.9858; 25.4649] -- -- 0.00  
## Subgroup = Approach bias 4 1.2683 [1.0702; 1.5031] 0 0 1.93  
## Subgroup = Attentional bias 2 0.6778 [0.2464; 1.8643] 0 0 0.05  
## Subgroup = Inhibitory bias 4 1.2208 [0.7529; 1.9798] 0.0972 0.3117 5.33  
## I^2  
## Subgroup = Executive functions --  
## Subgroup = Approach bias 0.0%  
## Subgroup = Attentional bias 0.0%  
## Subgroup = Inhibitory bias 43.7%  
##   
## Test for subgroup differences (random effects model):  
## Q d.f. p-value  
## Between groups 8.51 3 0.0366  
##   
## Details of meta-analysis methods:  
## - Inverse variance method  
## - Restricted maximum-likelihood estimator for tau^2  
## - Q-Profile method for confidence interval of tau^2 and tau  
## - Calculation of I^2 based on Q

It is now possible to extract the test statistics for the overall and subgroup effects, which can be used for the forest plot.

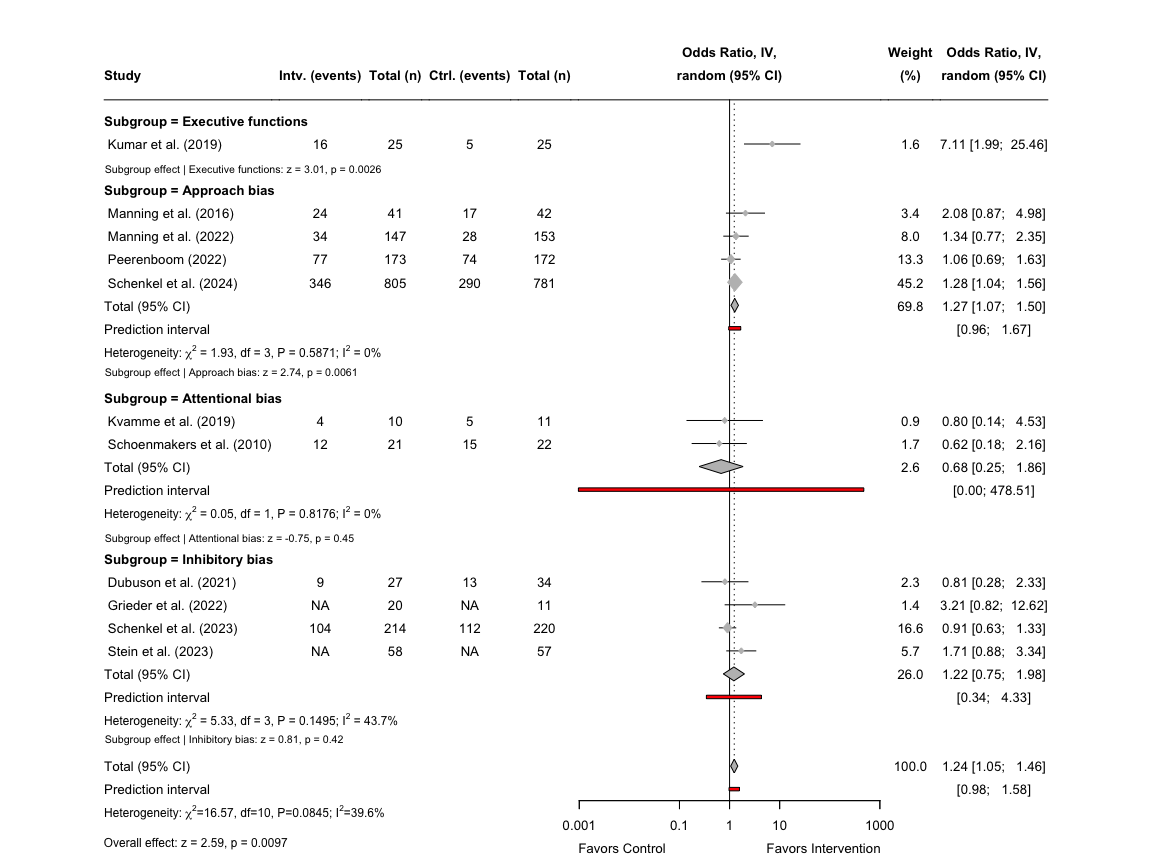
# Extract the test statistics for the overall effect and create a data frame  
ca\_sec\_zp\_overall\_3m <- data.frame(  
 lnOR = ca\_sec\_metagen\_3m$TE.random,  
 lnOR\_se = ca\_sec\_metagen\_3m$seTE.random,  
 z\_value = ca\_sec\_metagen\_3m$statistic.random,  
 p\_value = ca\_sec\_metagen\_3m$pval.random  
)  
  
# Extract the test statistics for the subgroup analysis and create a data frame  
ca\_sec\_zp\_subgroup\_3m <- data.frame(  
 subgroup = ca\_sec\_metagen\_3m$bylevs,  
 lnOR = ca\_sec\_metagen\_3m$TE.random.w,  
 lnOR\_se = ca\_sec\_metagen\_3m$seTE.random.w,  
 z\_value = ca\_sec\_metagen\_3m$statistic.random.w,  
 p\_value = ca\_sec\_metagen\_3m$pval.random.w  
)

For an easier way to add the individual effect sizes for each subgroup in the final forest plot, we can create a split of the data frame.

# Split the previously defined data frame by each subordinate domain  
ca\_sec\_subsplit\_3m <- split(ca\_sec\_zp\_subgroup\_3m, ca\_sec\_zp\_subgroup\_3m$subgroup)

Now that we have created a list for the test statistics for each of the subordinate domains, we can create the final forest plot.

# Now plot the forest plot again with new subgroup names  
forest(ca\_sec\_metagen\_3m,  
 prediction = TRUE,  
 prediction.subgroup = TRUE,  
 layout = "BMJ",  
 subgroup = TRUE,  
 overall = TRUE,  
 random = TRUE,  
 common = FALSE,  
 test.subgroup = FALSE,  
 print.stat = FALSE,  
 print.I2 = TRUE,  
 print.Q = TRUE,  
 print.pval.Q = TRUE,  
 print.tau2 = FALSE,  
 label.left = "Favors Control",  
 label.right = "Favors Intervention",  
 spacing = 1.2,  
 fontsize = 10,  
 digits = 2,  
 leftcols = c("studlab", "e\_abs", "e\_n", "c\_abs", "c\_n"),  
 leftlabs = c("Study", "Intv. (events)", "Total (n)",   
 "Ctrl. (events)", "Total (n)"),  
 just.addcols = "center"  
 )  
  
# Add a footer for the test statistics of the effects for executive functions  
footer\_sec\_exft\_ca\_3m <- paste0(  
 "Subgroup effect | ",  
 paste0(  
 ca\_sec\_subsplit\_3m$`Executive functions`,  
 ": z = ", round(ca\_sec\_subsplit\_3m$`Executive functions`$z\_value, 2),  
 ", p = ", formatC(ca\_sec\_subsplit\_3m$`Executive functions`$p\_value, digits = 2)  
 )  
)  
  
# Add a footer for the test statistics of the effects for approach bias  
footer\_sec\_apbm\_ca\_3m <- paste0(  
 "Subgroup effect | ",  
 paste0(  
 ca\_sec\_subsplit\_3m$`Approach bias`,  
 ": z = ", round(ca\_sec\_subsplit\_3m$`Approach bias`$z\_value, 2),  
 ", p = ", formatC(ca\_sec\_subsplit\_3m$`Approach bias`$p\_value, digits = 2)  
 )  
)  
  
# Add a footer for the test statistics of the effects for attentional bias  
footer\_sec\_atbm\_ca\_3m <- paste0(  
 "Subgroup effect | ",  
 paste0(  
 ca\_sec\_subsplit\_3m$`Attentional bias`,  
 ": z = ", round(ca\_sec\_subsplit\_3m$`Attentional bias`$z\_value, 2),  
 ", p = ", formatC(ca\_sec\_subsplit\_3m$`Attentional bias`$p\_value, digits = 2)  
 )  
)  
  
# Add a footer for the test statistics of the effects for inhibitory bias  
footer\_sec\_itbm\_ca\_3m <- paste0(  
 "Subgroup effect | ",  
 paste0(  
 ca\_sec\_subsplit\_3m$`Inhibitory bias`,  
 ": z = ", round(ca\_sec\_subsplit\_3m$`Inhibitory bias`$z\_value, 2),  
 ", p = ", formatC(ca\_sec\_subsplit\_3m$`Inhibitory bias`$p\_value, digits = 2)  
 )  
)  
  
# Add a footer combining the test statistics for the overall effect  
footer\_sec\_over\_ca\_3m <- paste0(  
 "Overall effect: ",  
 paste0(  
 "z = ", round(ca\_sec\_zp\_overall\_3m$z\_value, 2),  
 ", p = ", formatC(ca\_sec\_zp\_overall\_3m$p\_value, digits = 2),  
 collapse = "; "  
 )  
)  
  
# Adjust the footer for the executive functions right below the subgroup  
grid.text(footer\_sec\_exft\_ca\_3m, x = 0.091, y = 0.805,  
 just = "left", gp = gpar(fontsize = 8))  
  
# Adjust the footer for the approach bias right below the subgroup  
grid.text(footer\_sec\_apbm\_ca\_3m, x = 0.091, y = 0.57,  
 just = "left", gp = gpar(fontsize = 8))  
  
# Adjust the footer for the attentional bias right below the subgroup  
grid.text(footer\_sec\_atbm\_ca\_3m, x = 0.091, y = 0.378,  
 just = "left", gp = gpar(fontsize = 8))  
  
# Adjust the footer for the inhibitory bias right below the subgroup  
grid.text(footer\_sec\_itbm\_ca\_3m, x = 0.091, y = 0.145,  
 just = "left", gp = gpar(fontsize = 8))  
  
# Adjust the footer for the overall effect to the bottom left  
grid.text(footer\_sec\_over\_ca\_3m, x = 0.090, y = 0.025,  
 just = "left", gp = gpar(fontsize = 9))



1. Forest plot of the effects of individual subordinate domains on abstinence

## 3.2 Short-term effects on alcohol reduction by subgroups

As only approach bias modification programs were used among the identified studies, we do not need to re-run the analyses using a new subgroup parameter, and we can simply use the results in section 2.2. However, to add the option for applying subgroup labels or to quicly re-run the analyses with potentially new studies, we paste the procedure from section 2.2.

### 3.2.1 Meta-analysis on alcohol reduction

It is now possible to run the meta-analysis for alcohol reduction using the metagen() function, setting the Hedges *g* correction for the SMD and REML for the heterogeneity analysis.

# Run the meta-analysis stratified by superordinate domains and using the new labels  
red\_sec\_metagen\_3m <- metagen(TE = SMD,  
 seTE = SE,  
 studlab = study,  
 data = red\_d\_full\_3m,  
 sm = "SMD",  
 method.smd = "Hedges",  
 method.tau = "REML",  
 common = FALSE,  
 random = TRUE,  
 method.random.ci = "classic")  
  
# Print the results of the meta-analysis  
print(red\_sec\_metagen\_3m)

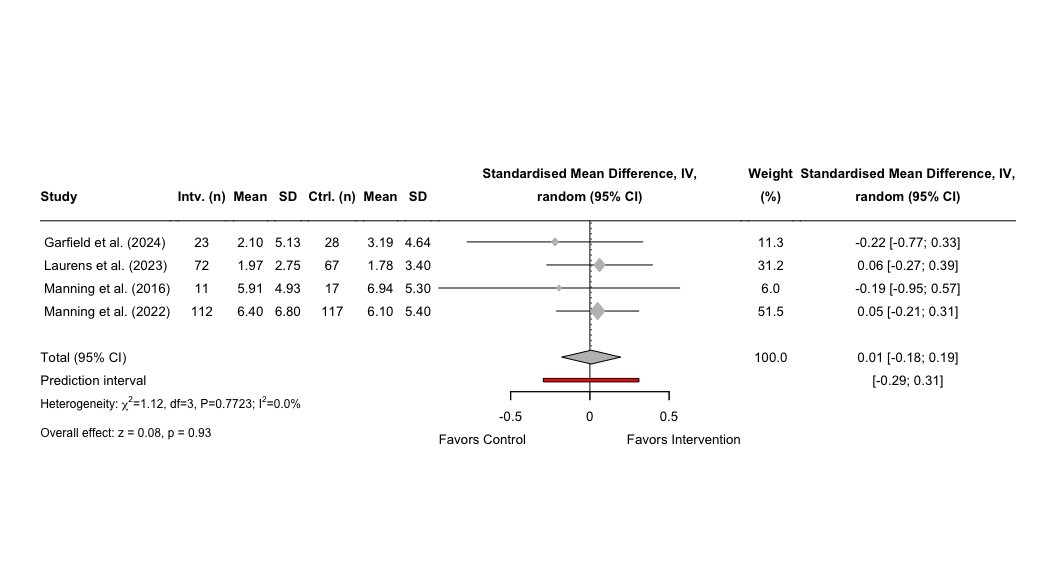
## Number of studies: k = 4  
##   
## SMD 95%-CI z p-value  
## Random effects model 0.0078 [-0.1782; 0.1938] 0.08 0.9345  
##   
## Quantifying heterogeneity (with 95%-CIs):  
## tau^2 = 0 [0.0000; 0.2560]; tau = 0 [0.0000; 0.5059]  
## I^2 = 0.0% [0.0%; 84.7%]; H = 1.00 [1.00; 2.56]  
##   
## Test of heterogeneity:  
## Q d.f. p-value  
## 1.12 3 0.7723  
##   
## Details of meta-analysis methods:  
## - Inverse variance method  
## - Restricted maximum-likelihood estimator for tau^2  
## - Q-Profile method for confidence interval of tau^2 and tau  
## - Calculation of I^2 based on Q

The test statistics can now be extracted from the meta-analysis and stored as a data frame for the forest plot.

# Extract the test statistics for the overall effect and create a data frame  
red\_zp\_over\_3m <- data.frame(  
 SMD = red\_sec\_metagen\_3m$TE.random,  
 SE = red\_sec\_metagen\_3m$seTE.random,  
 z\_value = red\_sec\_metagen\_3m$statistic.random,  
 p\_value = red\_sec\_metagen\_3m$pval.random  
)

With the test statistics stored as a data frame, we now turn to creating the forest plot, which follows the same design as previously used.

# Now plot the forest plot again with new subgroup names  
forest(red\_sec\_metagen\_3m,  
 prediction = TRUE,  
 layout = "BMJ",  
 overall = TRUE,  
 random = TRUE,  
 common = FALSE,  
 print.stat = FALSE,  
 print.I2 = TRUE,  
 print.Q = TRUE,  
 print.pval.Q = TRUE,  
 print.tau2 = FALSE,  
 label.left = "Favors Control",  
 label.right = "Favors Intervention",  
 spacing = 1.2,  
 fontsize = 10,  
 digits = 2,  
 leftcols = c("studlab", "e\_n", "e\_reduc\_mean", "e\_reduc\_sd", "c\_n", "c\_reduc\_mean",  
 "c\_reduc\_sd"),  
 leftlabs = c("Study", "Intv. (n)", "Mean", "SD",   
 "Ctrl. (n)", "Mean", "SD"),  
 just.addcols = "center"  
 )  
  
# Add a footer combining the test statistics for the overall effect  
footer\_sec\_over\_red\_3m <- paste0(  
 "Overall effect: ",  
 paste0(  
 "z = ", round(red\_zp\_over\_3m$z\_value, 2),  
 ", p = ", formatC(red\_zp\_over\_3m$p\_value, digits = 2),  
 collapse = "; "  
 )  
)  
  
# Adjust the footer for the overall effect to the bottom left  
grid.text(footer\_sec\_over\_red\_3m, x = 0.0385, y = 0.25,  
 just = "left", gp = gpar(fontsize = 9))



1. Forest plot of the effects of ApBM on alcohol reduction

## 3.3 Short-term effects on alcohol craving by subgroups

To explore the short-term effects on alcohol craving by each subordinate domain, we first need to assign new variable names to the subgroups to create a visually more appealing forest plot. We assign labels to three subordinate domains: Approach bias (Implicit cognition), Attentional bias (Implicit cognition), and Inhibitory bias (Implicit cognition).

# Assigning new variable names to the subgroups  
crav\_d\_full\_3m <- crav\_d\_full\_3m %>%  
 dplyr::rename(Subgroup = cog\_domain) %>%  
 dplyr::mutate(  
 Subgroup = factor(  
 Subgroup,  
 levels = c("Approach bias", "Attentional bias", "Inhibitory bias")  
 )  
 )

### 3.3.1 Meta-analysis on alcohol craving

Now with the newly defined subgroup labels, we can conduct the meta-analysis using metagen(), but this time set the parameter for subgroups to the label “Subgroup” and use the classic *z*-test for the random effects model.

# Run the meta-analysis stratified by superordinate domains and using the new labels  
crav\_sec\_metagen\_3m <- metagen(TE = SMD,  
 seTE = SE,  
 studlab = study,  
 data = crav\_d\_full\_3m,  
 subgroup = Subgroup,  
 sm = "SMD",  
 method.tau = "REML",  
 common = FALSE,  
 random = TRUE,  
 method.random.ci = "classic")  
  
# Print the results of the meta-analysis  
print(crav\_sec\_metagen\_3m)

## Number of studies: k = 9  
##   
## SMD 95%-CI z p-value  
## Random effects model -0.0992 [-0.3703; 0.1720] -0.72 0.4734  
##   
## Quantifying heterogeneity (with 95%-CIs):  
## tau^2 = 0.1089 [0.0200; 0.5794]; tau = 0.3300 [0.1414; 0.7612]  
## I^2 = 69.6% [39.5%; 84.8%]; H = 1.81 [1.29; 2.56]  
##   
## Test of heterogeneity:  
## Q d.f. p-value  
## 26.35 8 0.0009  
##   
## Results for subgroups (random effects model):  
## k SMD 95%-CI tau^2 tau Q  
## Subgroup = Approach bias 4 -0.1015 [-0.6085; 0.4054] 0.2291 0.4787 21.10  
## Subgroup = Attentional bias 3 -0.0164 [-0.4063; 0.3735] <0.0001 0.0021 2.58  
## Subgroup = Inhibitory bias 2 -0.1558 [-0.6193; 0.3076] 0.0573 0.2394 2.03  
## I^2  
## Subgroup = Approach bias 85.8%  
## Subgroup = Attentional bias 22.4%  
## Subgroup = Inhibitory bias 50.7%  
##   
## Test for subgroup differences (random effects model):  
## Q d.f. p-value  
## Between groups 0.21 2 0.8994  
##   
## Details of meta-analysis methods:  
## - Inverse variance method  
## - Restricted maximum-likelihood estimator for tau^2  
## - Q-Profile method for confidence interval of tau^2 and tau  
## - Calculation of I^2 based on Q

It is now possible to extract the test statistics for the overall and subgroup effects, which can be used for the forest plot.

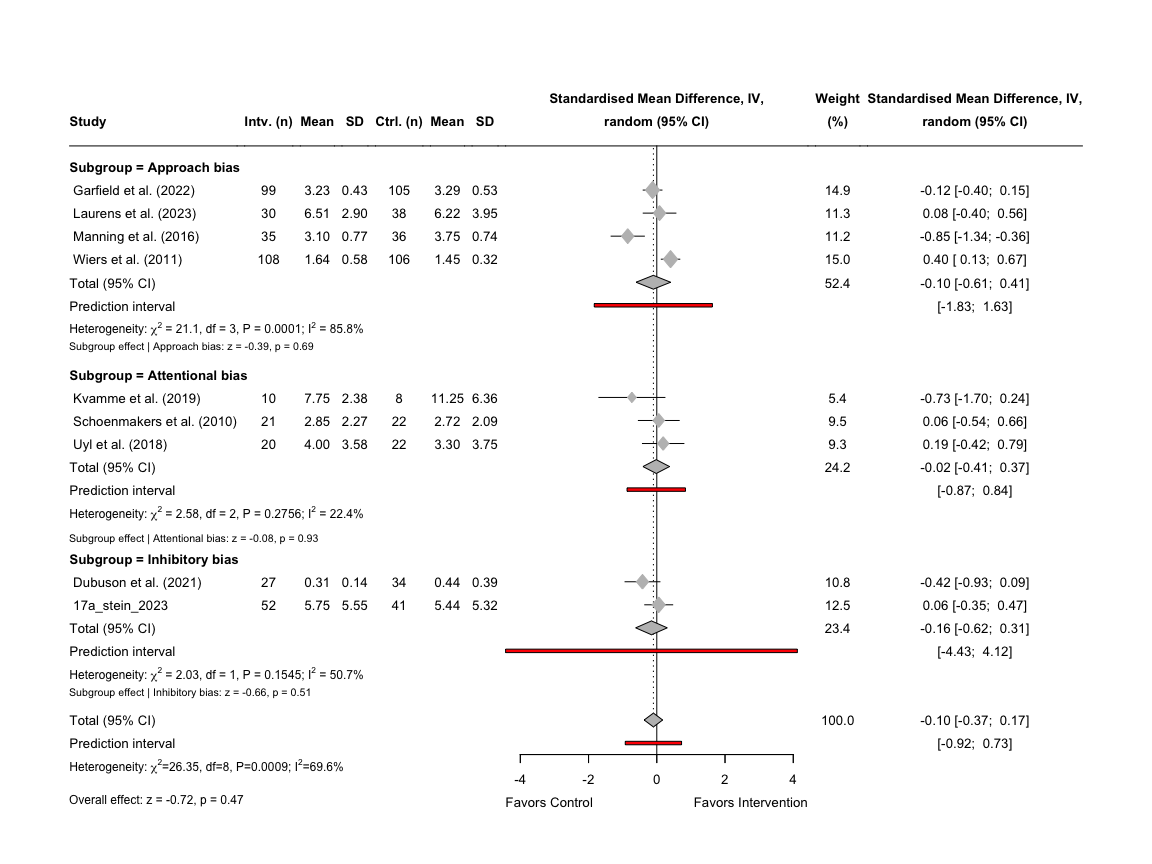
# Extract the test statistics for the overall effect and create a data frame  
crav\_sec\_zp\_overall\_3m <- data.frame(  
 SMD = crav\_sec\_metagen\_3m$TE.random,  
 SE = crav\_sec\_metagen\_3m$seTE.random,  
 z\_value = crav\_sec\_metagen\_3m$statistic.random,  
 p\_value = crav\_sec\_metagen\_3m$pval.random  
)  
  
# Extract the test statistics for the subgroup analysis and create a data frame  
crav\_sec\_zp\_subgroup\_3m <- data.frame(  
 subgroup = crav\_sec\_metagen\_3m$bylevs,  
 SMD = crav\_sec\_metagen\_3m$TE.random.w,  
 SE = crav\_sec\_metagen\_3m$seTE.random.w,  
 z\_value = crav\_sec\_metagen\_3m$statistic.random.w,  
 p\_value = crav\_sec\_metagen\_3m$pval.random.w  
)

For an easier way to add the individual effect sizes for each subgroup in the final forest plot, we can create a split of the data frame.

# Split the previously defined data frame by each subordinate domain  
crav\_sec\_subsplit\_3m <- split(crav\_sec\_zp\_subgroup\_3m, crav\_sec\_zp\_subgroup\_3m$subgroup)

Now that we have created a list for the test statistics for each of the subordinate domains, we can create the final forest plot.

# Now plot the forest plot again with new subgroup names  
forest(crav\_sec\_metagen\_3m,  
 prediction = TRUE,  
 prediction.subgroup = TRUE,  
 layout = "BMJ",  
 subgroup = TRUE,  
 overall = TRUE,  
 random = TRUE,  
 common = FALSE,  
 test.subgroup = FALSE,  
 print.stat = FALSE,  
 print.I2 = TRUE,  
 print.Q = TRUE,  
 print.pval.Q = TRUE,  
 print.tau2 = FALSE,  
 label.left = "Favors Control",  
 label.right = "Favors Intervention",  
 spacing = 1.2,  
 fontsize = 10,  
 digits = 2,  
 leftcols = c("studlab", "e\_n", "e\_crav\_mean", "e\_crav\_sd", "c\_n", "c\_crav\_mean",  
 "c\_crav\_sd"),  
 leftlabs = c("Study", "Intv. (n)", "Mean", "SD",   
 "Ctrl. (n)", "Mean", "SD"),  
 just.addcols = "center"  
 )  
  
# Add a footer for the test statistics of the effects for approach bias  
footer\_sec\_apbm\_crav\_3m <- paste0(  
 "Subgroup effect | ",  
 paste0(  
 crav\_sec\_subsplit\_3m$`Approach bias`,  
 ": z = ", round(crav\_sec\_subsplit\_3m$`Approach bias`$z\_value, 2),  
 ", p = ", formatC(crav\_sec\_subsplit\_3m$`Approach bias`$p\_value, digits = 2)  
 )  
)  
  
# Add a footer for the test statistics of the effects for attentional bias  
footer\_sec\_atbm\_crav\_3m <- paste0(  
 "Subgroup effect | ",  
 paste0(  
 crav\_sec\_subsplit\_3m$`Attentional bias`,  
 ": z = ", round(crav\_sec\_subsplit\_3m$`Attentional bias`$z\_value, 2),  
 ", p = ", formatC(crav\_sec\_subsplit\_3m$`Attentional bias`$p\_value, digits = 2)  
 )  
)  
  
# Add a footer for the test statistics of the effects for inhibitory bias  
footer\_sec\_itbm\_crav\_3m <- paste0(  
 "Subgroup effect | ",  
 paste0(  
 crav\_sec\_subsplit\_3m$`Inhibitory bias`,  
 ": z = ", round(crav\_sec\_subsplit\_3m$`Inhibitory bias`$z\_value, 2),  
 ", p = ", formatC(crav\_sec\_subsplit\_3m$`Inhibitory bias`$p\_value, digits = 2)  
 )  
)  
  
# Add a footer combining the test statistics for the overall effect  
footer\_sec\_over\_crav\_3m <- paste0(  
 "Overall effect: ",  
 paste0(  
 "z = ", round(crav\_sec\_zp\_overall\_3m$z\_value, 2),  
 ", p = ", formatC(crav\_sec\_zp\_overall\_3m$p\_value, digits = 2),  
 collapse = "; "  
 )  
)  
  
# Adjust the footer for the approach bias right below the subgroup  
grid.text(footer\_sec\_apbm\_crav\_3m, x = 0.06, y = 0.6,  
 just = "left", gp = gpar(fontsize = 8))  
  
# Adjust the footer for the attentional bias right below the subgroup  
grid.text(footer\_sec\_atbm\_crav\_3m, x = 0.06, y = 0.378,  
 just = "left", gp = gpar(fontsize = 8))  
  
# Adjust the footer for the inhibitory bias right below the subgroup  
grid.text(footer\_sec\_itbm\_crav\_3m, x = 0.06, y = 0.20,  
 just = "left", gp = gpar(fontsize = 8))  
  
# Adjust the footer for the overall effect to the bottom left  
grid.text(footer\_sec\_over\_crav\_3m, x = 0.06, y = 0.075,  
 just = "left", gp = gpar(fontsize = 9))



1. Forest plot of the effects of individual subordinate domains on craving

# 4. Tertiary analyses on intermediate and long-term effects (6 and 12 months)

The final part of the analyses will be conducted on the intermediate and long-term time points to explore for potential differential effects of cognitive training over time and to see how the effects compare to the results for the 0-3 month time point. Intermediate effects will consist of studies examining cognitive training effects six months after intervention/discharge, whereas the analyses on the long-term effects will consist of studies exploring the effects 12 months after discharge. We still use the excel-dataset “Extracted Clinical Outcome Data”, but this time we will only be filtering the data sheets by time point for the calculation of effect sizes and not when conducting the meta-analyses

## 4.1 Intermediate and long-term effects on continuous abstinence

The first part of our tertiary analyses will be conducted on our primary outcome of CA, and we calculate the effect sizes for both the intermediate and long-term time point.

### 4.1.1 Calculation of effect sizes for intermediate continuous and categorical variables

We want to follow the procedure of converting the continuous variables to categorical outcomes and vice versa as mentioned in section 2.1.2 and 2.1.3, and to begin with we only do this for the intermediate time point.

# Filter 6 month data first and store it as a new subset  
ca\_d\_6m <- ca\_d %>% filter(time\_point == "6 months")

The calculation of effect sizes will be done by the inbuilt functions in the meta-package, so for the categorical variables, we will use the metabin() function.

# Calculating the lnORs using metabin()  
ca\_metabin\_6m <- metabin(  
 event.e = e\_abs,  
 n.e = e\_n,  
 event.c = c\_abs,  
 n.c = c\_n,  
 studlab = study,  
 data = ca\_d\_6m,  
 sm = "OR",  
 method.tau = "REML",  
 method.random.ci = "classic",  
 random = TRUE,  
 incr = 0.5,  
 allstudies = TRUE  
)

After we have calculated the lnORs, we can create a new data frame and store these values alongside with the other relevant variables (e.g., time point, number of abstinents, etc.) from our original dataset. As performed previously, we calculate the ORs and SMDs using the Chinn-factor. Finally, we add the z- and p-values to the data-frame.

# Fetching the previously stored data frame and adding the lnORs, their SE and 95% CI  
ca\_d\_full\_6m <- data.frame(  
 row\_id = ca\_d\_6m$row\_id,  
 study = ca\_metabin\_6m$studlab,  
 time\_point = ca\_d\_6m$time\_point,  
 super\_domain = ca\_d\_6m$super\_domain,  
 cog\_domain = ca\_d\_6m$cog\_domain,  
 total\_n = ca\_d\_6m$e\_n + ca\_d\_6m$c\_n,  
 c\_n = ca\_d\_6m$c\_n,  
 c\_abs = ca\_d\_6m$c\_abs,  
 c\_non\_abs = ca\_d\_6m$c\_n - ca\_d\_6m$c\_abs,  
 e\_n = ca\_d\_6m$e\_n,  
 e\_abs = ca\_d\_6m$e\_abs,  
 e\_non\_abs = ca\_d\_6m$e\_n - ca\_d\_6m$e\_abs,  
# Extracting the lnORs from the metabin results  
 lnOR = ca\_metabin\_6m$TE,  
 SElnOR = ca\_metabin\_6m$seTE,  
 lnLCI = ca\_metabin\_6m$lower,  
 lnUCI = ca\_metabin\_6m$upper,  
# Calculating the ORs from the metabin results  
 OR = exp(ca\_metabin\_6m$TE),  
 OrSE = exp(ca\_metabin\_6m$TE) \* ca\_metabin\_6m$seTE,  
 OrLCI = exp(ca\_metabin\_6m$lower),  
 OrUCI = exp(ca\_metabin\_6m$upper),  
# Caculating the SMD, SE and 95% CI using the Chinn-function  
 SMD = ca\_metabin\_6m$TE \* chinn\_lnor\_to\_smd,  
 SE = ca\_metabin\_6m$seTE \* chinn\_lnor\_to\_smd,  
 LCI = ca\_metabin\_6m$lower \* chinn\_lnor\_to\_smd,  
 UCI = ca\_metabin\_6m$upper \* chinn\_lnor\_to\_smd,  
# Adding the z- and p-values  
 `z-value` = ca\_metabin\_6m$TE / ca\_metabin\_6m$seTE,  
 `p-value` = 2 \* (1 - pnorm(abs(ca\_metabin\_6m$TE / ca\_metabin\_6m$seTE))),  
# Adding a final note on how data was converted   
 Notes = "Continuous variables converted from categorical outcome (metabin) using Chinn's formula",  
 check.names = FALSE  
)

### 4.1.2 Calculation of effect sizes for long-term continuous and categorical variables

We can now apply the above methodology for our time point of 12 months, which requires us first to filter the data set.

# Filter 12 month data first and store it as a new subset  
ca\_d\_12m <- ca\_d %>% filter(time\_point == "12 months")

The calculation of effect sizes will be done by the inbuilt functions in the meta-package, so for the categorical variables, we will use the metabin() function.

# Calculating the lnORs using metabin()  
ca\_metabin\_12m <- metabin(  
 event.e = e\_abs,  
 n.e = e\_n,  
 event.c = c\_abs,  
 n.c = c\_n,  
 studlab = study,  
 data = ca\_d\_12m,  
 sm = "OR",  
 method.tau = "REML",  
 method.random.ci = "classic",  
 random = TRUE,  
 incr = 0.5,  
 allstudies = TRUE  
)

After we have calculated the lnORs, we can create a new data frame and store these values alongside with the other relevant variables (e.g., time point, number of abstinents, etc.) from our original dataset. As performed previously, we calculate the ORs and SMDs using the Chinn-factor. Finally, we add the z- and p-values to the data-frame.

# Fetching the previously stored data frame and adding the lnORs, their SE and 95% CI  
ca\_d\_full\_12m <- data.frame(  
 row\_id = ca\_d\_12m$row\_id,  
 study = ca\_metabin\_12m$studlab,  
 time\_point = ca\_d\_12m$time\_point,  
 super\_domain = ca\_d\_12m$super\_domain,  
 cog\_domain = ca\_d\_12m$cog\_domain,  
 total\_n = ca\_d\_12m$e\_n + ca\_d\_12m$c\_n,  
 c\_n = ca\_d\_12m$c\_n,  
 c\_abs = ca\_d\_12m$c\_abs,  
 c\_non\_abs = ca\_d\_12m$c\_n - ca\_d\_12m$c\_abs,  
 e\_n = ca\_d\_12m$e\_n,  
 e\_abs = ca\_d\_12m$e\_abs,  
 e\_non\_abs = ca\_d\_12m$e\_n - ca\_d\_12m$e\_abs,  
# Extracting the lnORs from the metabin results  
 lnOR = ca\_metabin\_12m$TE,  
 SElnOR = ca\_metabin\_12m$seTE,  
 lnLCI = ca\_metabin\_12m$lower,  
 lnUCI = ca\_metabin\_12m$upper,  
# Calculating the ORs from the metabin results  
 OR = exp(ca\_metabin\_12m$TE),  
 OrSE = exp(ca\_metabin\_12m$TE) \* ca\_metabin\_12m$seTE,  
 OrLCI = exp(ca\_metabin\_12m$lower),  
 OrUCI = exp(ca\_metabin\_12m$upper),  
# Caculating the SMD, SE and 95% CI using the Chinn-function  
 SMD = ca\_metabin\_12m$TE \* chinn\_lnor\_to\_smd,  
 SE = ca\_metabin\_12m$seTE \* chinn\_lnor\_to\_smd,  
 LCI = ca\_metabin\_12m$lower \* chinn\_lnor\_to\_smd,  
 UCI = ca\_metabin\_12m$upper \* chinn\_lnor\_to\_smd,  
# Adding the z- and p-values  
 `z-value` = ca\_metabin\_12m$TE / ca\_metabin\_12m$seTE,  
 `p-value` = 2 \* (1 - pnorm(abs(ca\_metabin\_12m$TE / ca\_metabin\_12m$seTE))),  
# Adding a final note on how data was converted   
 Notes = "Continuous variables converted from categorical outcome (metabin) using Chinn's formula",  
 check.names = FALSE  
)

### 4.1.3 Integrating the data for continuous abstinence for all time points

First we need to assign old labels to the data frame for 0-3 months and ensure that the variables are numeric.

# Assigning old variable names to the subgroups  
ca\_d\_full\_3m <- ca\_d\_full\_3m %>%  
 dplyr::rename(cog\_domain = Subgroup) %>%  
 dplyr::mutate( # Changing the order and label for the two subgroups  
 super\_domain = factor(super\_domain,  
 levels = c("Explicit cognition", "Implicit cognition"),  
 labels = c("Explicit", "Implicit")))  
  
# Converting to numeric values   
ca\_d\_full\_3m <- ca\_d\_full\_3m %>%  
 dplyr::mutate(across(c(c\_n, e\_n, c\_abs, e\_abs, c\_non\_abs, e\_non\_abs, total\_n),  
 ~ as.numeric(.)))

We can now integrate the two data frames with the data frame for 0-3 months using the the bind\_rows function.

# Combine the three data frames for CA on all time points and round all the values 5 decimals  
ca\_d\_full\_0\_12m <- bind\_rows(ca\_d\_full\_3m, ca\_d\_full\_6m, ca\_d\_full\_12m) %>% mutate(across(where(is.numeric), ~ round(.x, 5))) %>%  
 # Arrange the studies by their pre-defined row-numbers  
 arrange(row\_id)

To use the same author labels in APA-format across all studies and sub-studies, we need to apply the previously defined study labels to our integrated data set.

# Assign the new study labels to the combined data frame  
ca\_d\_full\_0\_12m$study <- ifelse(ca\_d\_full\_0\_12m$study %in% names(study\_labels),  
 study\_labels[ca\_d\_full\_0\_12m$study],  
 ca\_d\_full\_0\_12m$study)

### 4.1.4 Meta-analysis on intermediate and long-term effects on continuous abstinence

To prepare for the final forest plot we need to make some adjustments to the integrated data frame such as assigning NAs to empty values. As we only have one study on explicit cognitive training, we will have to filter this study out, and proceed with the meta-analysis exclusively on implicit cognitive training.

# Assign NAs strings to studies with missing event data  
ca\_d\_imp\_0\_12m <- ca\_d\_full\_0\_12m %>%  
 mutate(across(c(e\_abs, e\_non\_abs, e\_n, c\_abs, c\_non\_abs, c\_n), ~ ifelse(is.na(.), "NA", as.character(.)))) %>%  
 filter(study != "Kumar et al. (2019)") # Filter out Kumar et al. (2019)

To make it clearer in the final forest plot, we assign new variable names to the stratification by time point, and sort them with the earliest time point on top and all the studies alphabetically.

# Assigning new variable names to the subgroups  
ca\_d\_imp\_0\_12m <- ca\_d\_imp\_0\_12m %>%  
 dplyr::rename("Time point" = time\_point) %>% # Renaming the time\_point variable  
 arrange(study) %>% # Arrange the studies alphabetically  
 arrange(desc(`Time point` == "0-3 months"), # Arrange then by Time Point  
 desc(`Time point` == "6 months"),  
 desc(`Time point` == "12 months")  
 )

Now with the newly defined subgroup labels, we can conduct the meta-analysis using metagen(), but this time set the parameter for subgroups to the label “Time point” and use the classic *z*-test for the random effects model.

# Run the meta-analysis stratified by superordinate domains and using the new labels  
ca\_ter\_metagen\_0\_12m <- metagen(TE = lnOR,  
 seTE = SElnOR,  
 studlab = study,  
 data = ca\_d\_imp\_0\_12m,  
 subgroup = `Time point`,  
 sm = "OR",  
 method.tau = "REML",  
 common = FALSE,  
 random = TRUE,  
 method.random.ci = "classic")  
  
# Print the results of the meta-analysis  
print(ca\_ter\_metagen\_0\_12m)

## Number of studies: k = 23  
##   
## OR 95%-CI z p-value  
## Random effects model 1.1860 [1.0863; 1.2948] 3.81 0.0001  
##   
## Quantifying heterogeneity (with 95%-CIs):  
## tau^2 < 0.0001 [0.0000; 0.1336]; tau = 0.0010 [0.0000; 0.3655]  
## I^2 = 4.6% [0.0%; 35.7%]; H = 1.02 [1.00; 1.25]  
##   
## Test of heterogeneity:  
## Q d.f. p-value  
## 23.06 22 0.3985  
##   
## Results for subgroups (random effects model):  
## k OR 95%-CI tau^2 tau Q I^2  
## Time point = 0-3 months 10 1.2103 [1.0450; 1.4017] <0.0001 0.0012 9.26 2.8%  
## Time point = 6 months 4 1.1497 [0.9448; 1.3991] 0 0 2.91 0.0%  
## Time point = 12 months 9 1.1956 [1.0105; 1.4145] 0.0139 0.1179 10.71 25.3%  
##   
## Test for subgroup differences (random effects model):  
## Q d.f. p-value  
## Between groups 0.17 2 0.9177  
##   
## Details of meta-analysis methods:  
## - Inverse variance method  
## - Restricted maximum-likelihood estimator for tau^2  
## - Q-Profile method for confidence interval of tau^2 and tau  
## - Calculation of I^2 based on Q

It is now possible to extract the test statistics for the individual time points, which can be used for the forest plot.

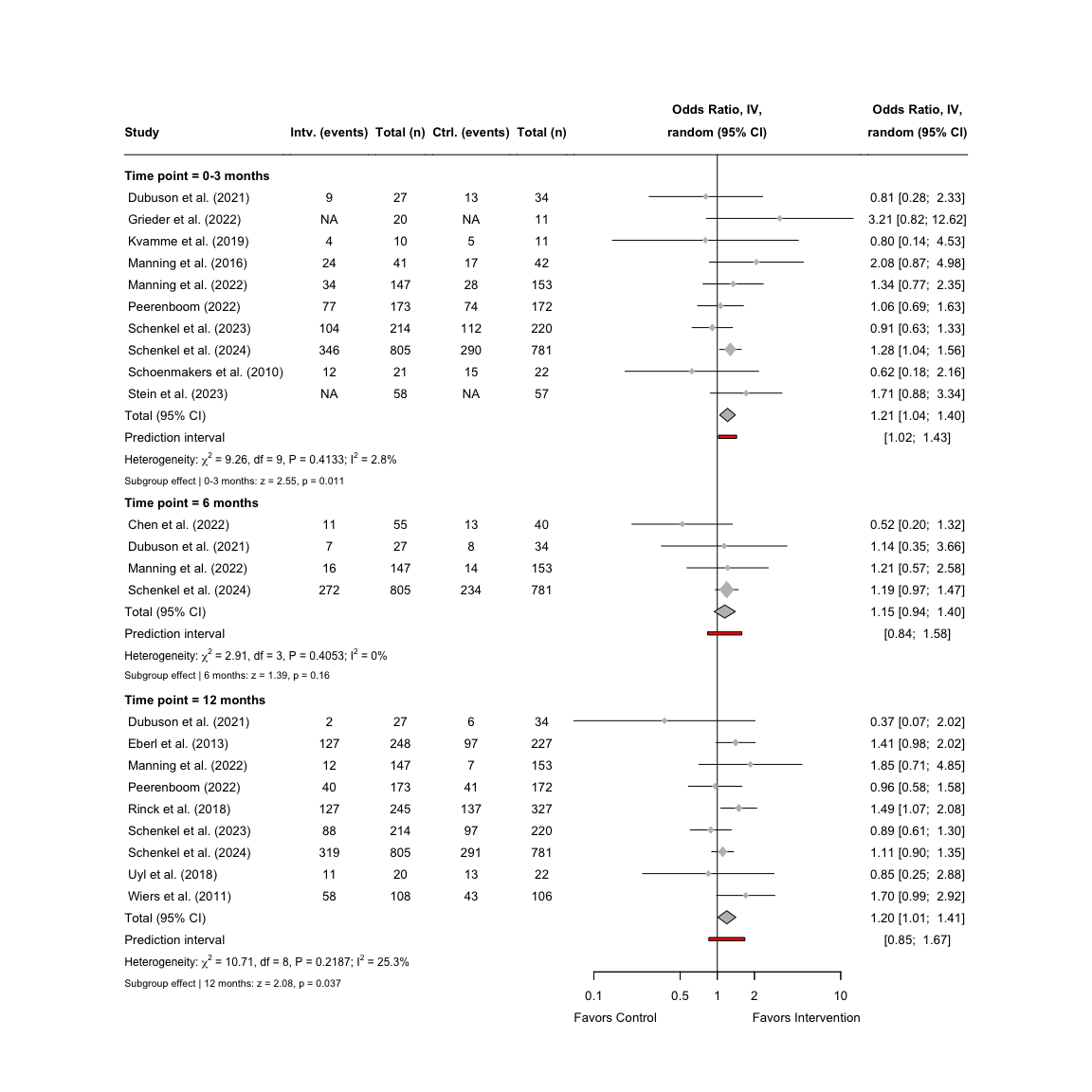
# Extract the test statistics for the subgroup analysis and create a data frame  
ca\_ter\_zp\_subgroup\_012m <- data.frame(  
 subgroup = ca\_ter\_metagen\_0\_12m$bylevs,  
 lnOR = ca\_ter\_metagen\_0\_12m$TE.random.w,  
 lnOR\_se = ca\_ter\_metagen\_0\_12m$seTE.random.w,  
 z\_value = ca\_ter\_metagen\_0\_12m$statistic.random.w,  
 p\_value = ca\_ter\_metagen\_0\_12m$pval.random.w  
)

For an easier way to add the individual effect sizes for each subgroup in the final forest plot, we can create a split of the data frame.

# Split the previously defined data frame by each subordinate domain  
ca\_ter\_subsplit\_012m <- split(ca\_ter\_zp\_subgroup\_012m, ca\_ter\_zp\_subgroup\_012m$subgroup)

Now that we have created a list for the test statistics for each of the time points, we can create the final forest plot.

# Now plot the forest plot again with new subgroup names  
forest(ca\_ter\_metagen\_0\_12m,  
 prediction = TRUE,  
 prediction.subgroup = TRUE,  
 layout = "BMJ",  
 subgroup = TRUE,  
 overall = FALSE,  
 random = TRUE,  
 common = FALSE,  
 test.subgroup = FALSE,  
 print.stat = FALSE,  
 print.I2 = TRUE,  
 print.Q = TRUE,  
 print.pval.Q = TRUE,  
 print.tau2 = FALSE,  
 label.left = "Favors Control",  
 label.right = "Favors Intervention",  
 spacing = 1.2,  
 fontsize = 10,  
 digits = 2,  
 leftcols = c("studlab", "e\_abs", "e\_n", "c\_abs", "c\_n"),  
 leftlabs = c("Study", "Intv. (events)", "Total (n)",   
 "Ctrl. (events)", "Total (n)"),  
 just.addcols = "center"  
 )  
  
# Add a footer for the test statistics of the effects after 0-3 months  
footer\_ter\_03\_ca <- paste0(  
 "Subgroup effect | ",  
 paste0(  
 ca\_ter\_subsplit\_012m$`0-3 months`,  
 ": z = ", round(ca\_ter\_subsplit\_012m$`0-3 months`$z\_value, 2),  
 ", p = ", formatC(ca\_ter\_subsplit\_012m$`0-3 months`$p\_value, digits = 2)  
 )  
)  
  
# Add a footer for the test statistics of the effects after 6 months  
footer\_ter\_6\_ca <- paste0(  
 "Subgroup effect | ",  
 paste0(  
 ca\_ter\_subsplit\_012m$`6 months`,  
 ": z = ", round(ca\_ter\_subsplit\_012m$`6 months`$z\_value, 2),  
 ", p = ", formatC(ca\_ter\_subsplit\_012m$`6 months`$p\_value, digits = 2)  
 )  
)  
  
# Add a footer for the test statistics of the effects after 12 months  
footer\_ter\_12\_ca <- paste0(  
 "Subgroup effect | ",  
 paste0(  
 ca\_ter\_subsplit\_012m$`12 months`,  
 ": z = ", round(ca\_ter\_subsplit\_012m$`12 months`$z\_value, 2),  
 ", p = ", formatC(ca\_ter\_subsplit\_012m$`12 months`$p\_value, digits = 2)  
 )  
)  
  
# Adjust the footer for the 0-3 month time point right below the subgroup  
grid.text(footer\_ter\_03\_ca, x = 0.114, y = 0.56,  
 just = "left", gp = gpar(fontsize = 8))  
  
  
# Adjust the footer for the 6 month time point right below the subgroup  
grid.text(footer\_ter\_6\_ca, x = 0.114, y = 0.3825,  
 just = "left", gp = gpar(fontsize = 8))  
  
# Cover up unwanted group heterogeneity statistics  
grid.rect(  
 x = 0.1,  
 y = 0.1,  
 width = 0.6,  
 height = 0.02,  
 gp = gpar(fill = "white", col = NA)  
)  
  
# Adjust the footer for the 12 month time point right below the subgroup  
grid.text(footer\_ter\_12\_ca, x = 0.114, y = 0.1,  
 just = "left", gp = gpar(fontsize = 8))



1. Forest plot for the short-, intermediate-, and long-term effects on abstinence

### 4.1.5 Publication bias for intermediate effects on continuous abstinence

To assess for potential publication bias across all time points, we need to create a funnel plot for the studies assessing the intermediate effects. Before the funnel plot can be created, we will be changing the author names to fit the APA-format using the previously defined study labels. Finally, we want to filter out the study by Kumar et al. (2019), as it is not included in the tertiary analyses.

# Assign the new study labels to the data frame  
ca\_d\_full\_6m$study <- ifelse(ca\_d\_full\_6m$study %in% names(study\_labels),  
 study\_labels[ca\_d\_full\_6m$study],  
 ca\_d\_full\_6m$study)  
  
# Filter out the study by Kumar et al. (2019)  
ca\_d\_imp\_6m <- ca\_d\_full\_6m %>%   
 filter(study != "Kumar et al. (2019)")

After defining the new labels, we can run a separate meta-analysis on the intermediate time point.

# Run the meta-analysis stratified by superordinate domains and using the new labels  
ca\_ter\_metagen\_6m <- metagen(TE = lnOR,  
 seTE = SElnOR,  
 studlab = study,  
 data = ca\_d\_imp\_6m,  
 sm = "OR",  
 method.tau = "REML",  
 common = FALSE,  
 random = TRUE)

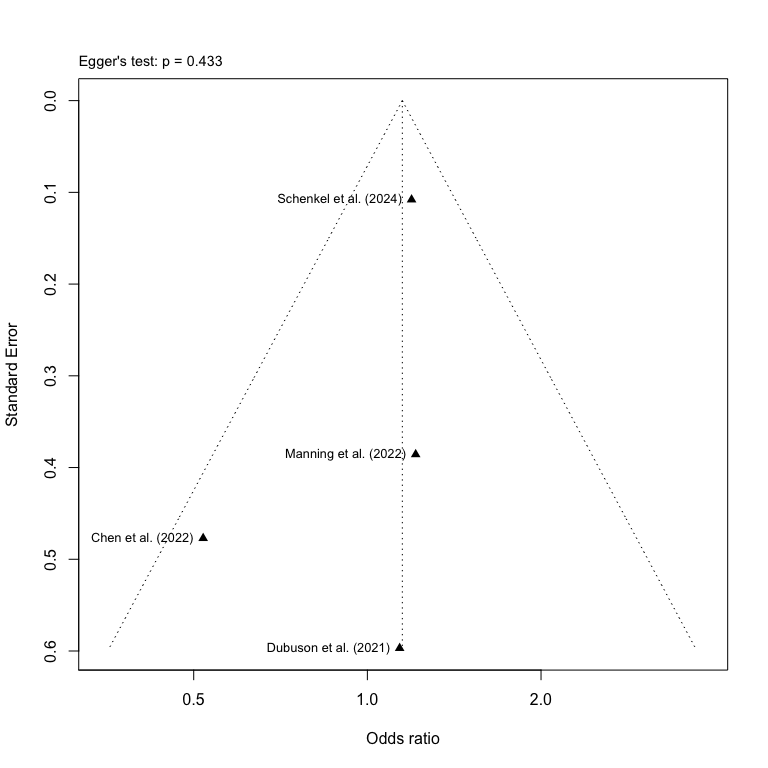
Now we can run an Egger’s test, but due to a low number of studies, we need to set the minimum *k* to four studies. The results will, however, be less reliable given the small number of studies, and interpretation of the results should be cautioned.

# Conduct an Egger's test using the previously calculated treatment effects  
ca\_egger\_6m <- metabias(ca\_ter\_metagen\_6m, method.bias = "Egger", k.min = 4)  
  
# Print the results of the Egger's test  
print(ca\_egger\_6m)

## Linear regression test of funnel plot asymmetry  
##   
## Test result: t = -0.97, df = 2, p-value = 0.4334  
## Bias estimate: -0.7764 (SE = 0.7984)  
##   
## Details:  
## - multiplicative residual heterogeneity variance (tau^2 = 0.9887)  
## - predictor: standard error  
## - weight: inverse variance  
## - reference: Egger et al. (1997), BMJ

Now with the results from the Egger’s test, we can now create the funnel plot using the meta-package inbuilt function funnel(). We can add the test statistic from the Egger’s test to the funnel plot using mtext(). As there is a high variability among these studies, we will need to adjust the limits of the x-axis so the studies can be better displayed.

# Create a funnel plot with adjusted x-axis  
funnel(ca\_ter\_metagen\_6m,  
 xlab = "Odds ratio",  
 studlab = TRUE,  
 pch = 17,  
 cex = 1)  
  
# Add Egger's test statistic to the plot  
ca\_egger\_text\_6m <- paste0("Egger's test: p = ",   
 round(ca\_egger\_6m$p.value, 3))  
  
# Define the actual placement of the statistics on the funnel plot  
mtext(ca\_egger\_text\_6m, side=3, line=0.5, adj=0, cex=0.9, col="black")



1. Funnel plot for studies with intermediate effects on abstinence

### 4.1.6 Publication bias for long-term effects on continuous abstinence

After we have created a funnel plot for the intermediate time point, we can now create a funnel plot for the studies assessing the long-term effects. Before the funnel plot can be created, we will be changing the author names to fit the APA-format using the previously defined study labels.

# Assign the new study labels to the data frame  
ca\_d\_full\_12m$study <- ifelse(ca\_d\_full\_12m$study %in% names(study\_labels),  
 study\_labels[ca\_d\_full\_12m$study],  
 ca\_d\_full\_12m$study)

After defining the new labels, we can run a separate meta-analysis on the long-term time point.

# Run the meta-analysis stratified by superordinate domains and using the new labels  
ca\_ter\_metagen\_12m <- metagen(TE = lnOR,  
 seTE = SElnOR,  
 studlab = study,  
 data = ca\_d\_full\_12m,  
 sm = "OR",  
 method.tau = "REML",  
 common = FALSE,  
 random = TRUE)

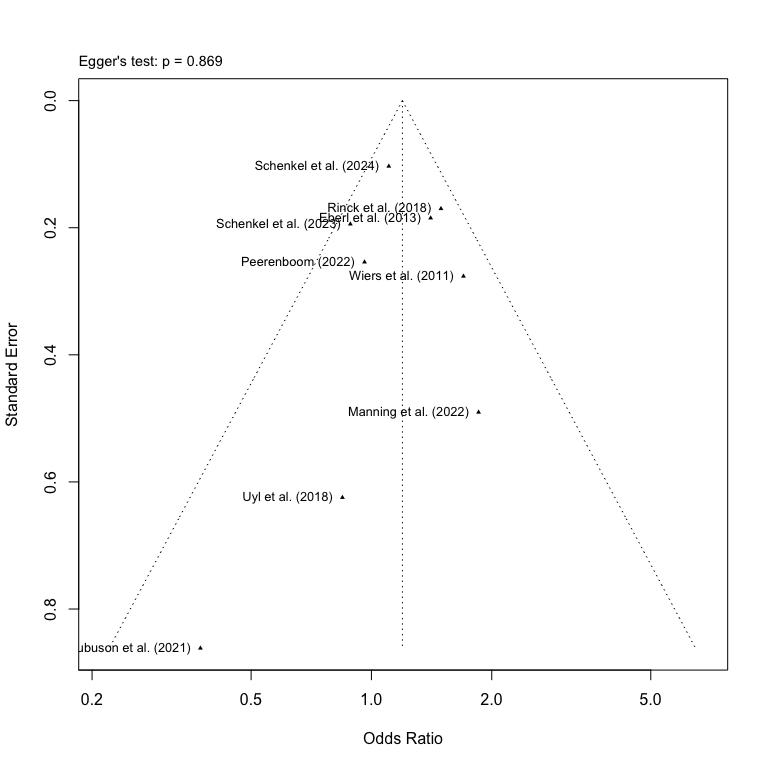
Now we can run an Egger’s test, but due to a low number of studies, we need to set the minimum *k* to eight studies. The results will, however, be less reliable given the small number of studies, and interpretation of the results should be cautioned.

# Conduct an Egger's test using the previously calculated treatment effects  
ca\_egger\_12m <- metabias(ca\_ter\_metagen\_12m, method.bias = "Egger", k.min = 8)  
  
# Print the results of the Egger's test  
print(ca\_egger\_12m)

## Linear regression test of funnel plot asymmetry  
##   
## Test result: t = -0.17, df = 7, p-value = 0.8694  
## Bias estimate: -0.1390 (SE = 0.8151)  
##   
## Details:  
## - multiplicative residual heterogeneity variance (tau^2 = 1.5237)  
## - predictor: standard error  
## - weight: inverse variance  
## - reference: Egger et al. (1997), BMJ

Now with the results from the Egger’s test, we can now create the funnel plot using the meta-package inbuilt function funnel(). We can add the test statistic from the Egger’s test to the funnel plot using mtext(). As there is a high variability among these studies, we will need to adjust the limits of the x-axis so the studies can be better displayed.

# Create a funnel plot for the 12 month time point  
funnel(ca\_ter\_metagen\_12m,  
 xlab = "Odds Ratio",  
 studlab = TRUE,  
 pch = 17,  
 cex = 0.5)  
  
# Add Egger's test statistic to the plot  
ca\_egger\_text\_12m <- paste0("Egger's test: p = ",   
 round(ca\_egger\_12m$p.value, 3))  
  
# Define the actual placement of the statistics on the funnel plot  
mtext(ca\_egger\_text\_12m, side=3, line=0.5, adj=0, cex=0.9, col="black")



1. Funnel plot of studies with long-term effects on abstinence

## 4.2 Intermediate and long-term effects on alcohol reduction

The second part of our tertiary analyses will be conducted on our secondary outcome of alcohol reduction, and we calculate the effect sizes for both the intermediate and long-term time point.

### 4.2.1 Calculation of effect sizes for intermediate continuous and categorical variables

We want to follow the procedure of as in section 4.1.1 and filter the data by the time point “6 months”.

# Filter 6 month data first and store it as a new subset  
red\_d\_6m <- red\_d %>% filter(time\_point == "6 months")

As we only have continuous data for alcohol reduction, we only need to run the metacont() on our dataset to calculate the SMDs using Hedge’s *g* correction.

# Calculating the SMDs (Hedge's g) using metacont()  
red\_metacont\_6m <- metacont(  
 n.e = e\_n,  
 mean.e = e\_reduc\_mean,  
 sd.e = e\_reduc\_sd,  
 n.c = c\_n,  
 mean.c = c\_reduc\_mean,  
 sd.c = c\_reduc\_sd,  
 studlab = study,  
 data = red\_d\_6m,  
 sm = "SMD",  
 method.smd = "Hedges",  
 method.tau = "REML",  
 method.random.ci = "classic",  
 random = TRUE  
)

Now we can combine the calculated effect sizes and add them to the original data set without changing the original variable names.

# Combine the SMDs calculated with original variables and store it in a new data frame  
red\_d\_full\_6m <- data.frame(  
 row\_id = red\_d\_6m$row\_id,  
 study = red\_metacont\_6m$studlab,  
 time\_point = red\_d\_6m$time\_point,  
 super\_domain = red\_d\_6m$super\_domain,  
 cog\_domain = red\_d\_6m$cog\_domain,  
 total\_n = red\_d\_6m$e\_n + red\_d\_6m$c\_n, # Calculating total number of participants  
 c\_n = red\_d\_6m$c\_n,  
 c\_reduc\_mean = red\_d\_6m$c\_reduc\_mean,  
 c\_reduc\_sd = red\_d\_6m$c\_reduc\_sd,  
 e\_n = red\_d\_6m$e\_n,  
 e\_reduc\_mean = red\_d\_6m$e\_reduc\_mean,  
 e\_reduc\_sd = red\_d\_6m$e\_reduc\_sd,  
 # Adding calculated SMDs to the dataframe   
 SMD = red\_metacont\_6m$TE,  
 SE = red\_metacont\_6m$seTE,  
 LCI = red\_metacont\_6m$lower,  
 UCI = red\_metacont\_6m$upper,  
 # Adding the z- and p-values  
 `z-value` = red\_metacont\_6m$TE / red\_metacont\_6m$seTE,  
 `p-value` = 2 \* (1 - pnorm(abs(red\_metacont\_6m$TE / red\_metacont\_6m$seTE))),  
 # Adding a final note on how data was converted   
 Notes = "Calculated using metacont()",  
 check.names = FALSE # Removing the period as a space  
)

### 4.2.2 Calculation of effect sizes for long-term continuous and categorical variables

We repeat the procedure as in the above section but this time filter the data by the time point “12 months”.

# Filter 6 month data first and store it as a new subset  
red\_d\_12m <- red\_d %>% filter(time\_point == "12 months")

As we only have continuous data for alcohol reduction, we only need to run the metacont() on our dataset to calculate the SMDs using Hedge’s *g* correction.

# Calculating the SMDs (Hedge's g) using metacont()  
red\_metacont\_12m <- metacont(  
 n.e = e\_n,  
 mean.e = e\_reduc\_mean,  
 sd.e = e\_reduc\_sd,  
 n.c = c\_n,  
 mean.c = c\_reduc\_mean,  
 sd.c = c\_reduc\_sd,  
 studlab = study,  
 data = red\_d\_12m,  
 sm = "SMD",  
 method.smd = "Hedges",  
 method.tau = "REML",  
 method.random.ci = "classic",  
 random = TRUE  
)

Now we can combine the calculated effect sizes and add them to the original data set without changing the original variable names.

# Combine the SMDs calculated with original variables and store it in a new data frame  
red\_d\_full\_12m <- data.frame(  
 row\_id = red\_d\_12m$row\_id,  
 study = red\_metacont\_12m$studlab,  
 time\_point = red\_d\_12m$time\_point,  
 super\_domain = red\_d\_12m$super\_domain,  
 cog\_domain = red\_d\_12m$cog\_domain,  
 total\_n = red\_d\_12m$e\_n + red\_d\_12m$c\_n, # Calculating total number of participants  
 c\_n = red\_d\_12m$c\_n,  
 c\_reduc\_mean = red\_d\_12m$c\_reduc\_mean,  
 c\_reduc\_sd = red\_d\_12m$c\_reduc\_sd,  
 e\_n = red\_d\_12m$e\_n,  
 e\_reduc\_mean = red\_d\_12m$e\_reduc\_mean,  
 e\_reduc\_sd = red\_d\_12m$e\_reduc\_sd,  
 # Adding calculated SMDs to the dataframe   
 SMD = red\_metacont\_12m$TE,  
 SE = red\_metacont\_12m$seTE,  
 LCI = red\_metacont\_12m$lower,  
 UCI = red\_metacont\_12m$upper,  
 # Adding the z- and p-values  
 `z-value` = red\_metacont\_12m$TE / red\_metacont\_12m$seTE,  
 `p-value` = 2 \* (1 - pnorm(abs(red\_metacont\_12m$TE / red\_metacont\_12m$seTE))),  
 # Adding a final note on how data was converted   
 Notes = "Calculated using metacont()",  
 check.names = FALSE # Removing the period as a space  
)

### 4.2.3 Integrating the data for alcohol reduction for all time points

We can now integrate the two data frames with the data frame for 0-3 months using the the bind\_rows function.

# Combine the three data frames for craving on all time points and round all the values 5 decimals  
red\_d\_full\_0\_12m <- bind\_rows(red\_d\_full\_3m, red\_d\_full\_6m, red\_d\_full\_12m) %>% mutate(across(where(is.numeric), ~ round(.x, 5))) %>%  
 # Arrange the studies by their pre-defined row-numbers  
 arrange(row\_id)

To use the same author labels in APA-format across all studies and sub-studies, we need to apply the previously defined study labels to our integrated data set.

# Assign the new study labels to the combined data frame  
red\_d\_full\_0\_12m$study <- ifelse(red\_d\_full\_0\_12m$study %in% names(study\_labels),  
 study\_labels[red\_d\_full\_0\_12m$study],  
 red\_d\_full\_0\_12m$study)

### 4.2.4 Meta-analysis on intermediate and long-term alcohol reduction

To make it clearer in the final forest plot, we assign new variable names to the stratification by time point, and sort them with the earliest time point on top.

# Assigning new variable names to the subgroups  
red\_d\_full\_0\_12m <- red\_d\_full\_0\_12m %>%  
 dplyr::rename("Time point" = time\_point) %>% # Renaming the time\_point variable  
 arrange(study) %>% # Arrange the studies alphabetically  
 arrange(desc(`Time point` == "0-3 months"), # Arrange then by Time Point  
 desc(`Time point` == "6 months"),  
 desc(`Time point` == "12 months")  
 )

Now with the newly defined subgroup labels, we can conduct the meta-analysis using metagen(), but this time set the parameter for subgroups to the label “Time point” and use the classic *z*-test for the random effects model.

# Run the meta-analysis stratified by superordinate domains and using the new labels  
red\_ter\_metagen\_0\_12m <- metagen(TE = SMD,  
 seTE = SE,  
 studlab = study,  
 data = red\_d\_full\_0\_12m,  
 subgroup = `Time point`,  
 sm = "SMD",  
 method.tau = "REML",  
 common = FALSE,  
 random = TRUE,  
 method.random.ci = "classic")  
  
# Print the results of the meta-analysis  
print(red\_ter\_metagen\_0\_12m)

## Number of studies: k = 8  
##   
## SMD 95%-CI z p-value  
## Random effects model 0.0745 [-0.0464; 0.1954] 1.21 0.2272  
##   
## Quantifying heterogeneity (with 95%-CIs):  
## tau^2 = 0 [0.0000; 0.0718]; tau = 0 [0.0000; 0.2679]  
## I^2 = 0.0% [0.0%; 67.6%]; H = 1.00 [1.00; 1.76]  
##   
## Test of heterogeneity:  
## Q d.f. p-value  
## 3.04 7 0.8809  
##   
## Results for subgroups (random effects model):  
## k SMD 95%-CI tau^2 tau Q I^2  
## Time point = 0-3 months 4 0.0078 [-0.1782; 0.1938] 0 0 1.12 0.0%  
## Time point = 6 months 3 0.1262 [-0.0699; 0.3224] 0 0 1.07 0.0%  
## Time point = 12 months 1 0.1177 [-0.1544; 0.3897] -- -- 0.00 --  
##   
## Test for subgroup differences (random effects model):  
## Q d.f. p-value  
## Between groups 0.86 2 0.6511  
##   
## Details of meta-analysis methods:  
## - Inverse variance method  
## - Restricted maximum-likelihood estimator for tau^2  
## - Q-Profile method for confidence interval of tau^2 and tau  
## - Calculation of I^2 based on Q

It is now possible to extract the test statistics for the individual time points, which can be used for the forest plot.

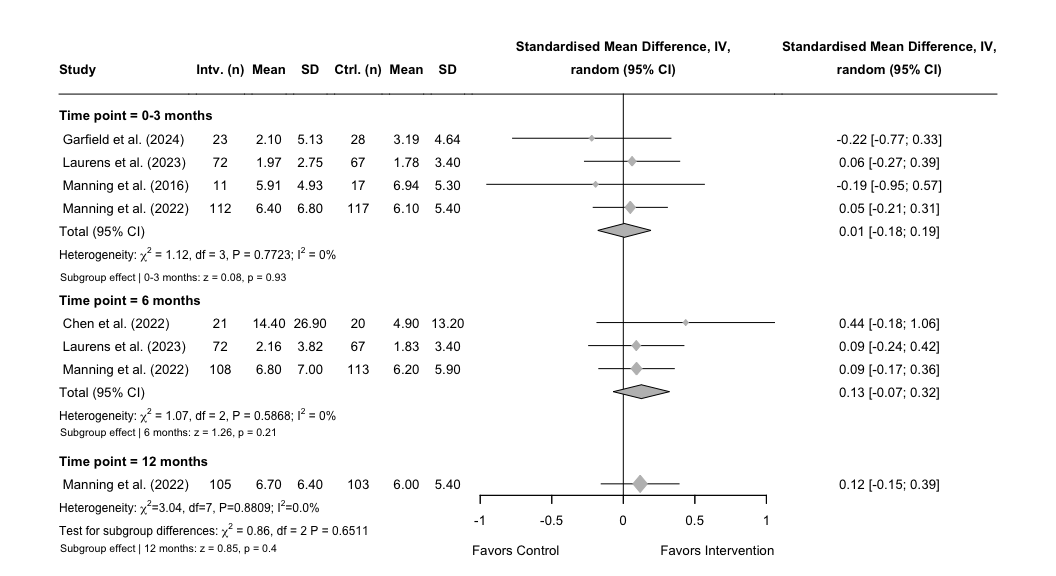
# Extract the test statistics for the subgroup analysis and create a data frame  
red\_ter\_zp\_subgroup\_012m <- data.frame(  
 subgroup = red\_ter\_metagen\_0\_12m$bylevs,  
 SMD = red\_ter\_metagen\_0\_12m$TE.random.w,  
 SE = red\_ter\_metagen\_0\_12m$seTE.random.w,  
 z\_value = red\_ter\_metagen\_0\_12m$statistic.random.w,  
 p\_value = red\_ter\_metagen\_0\_12m$pval.random.w  
)

For an easier way to add the individual effect sizes for each subgroup in the final forest plot, we can create a split of the data frame.

# Split the previously defined data frame by each subordinate domain  
red\_ter\_subsplit\_012m <- split(red\_ter\_zp\_subgroup\_012m, red\_ter\_zp\_subgroup\_012m$subgroup)

Now that we have created a list for the test statistics for each of the time points, we can create the final forest plot. We set the overall effect to FALSE, as we do not want to pool the effects across reports of different time points in the same study.

# Now plot the forest plot again with new subgroup names  
forest(red\_ter\_metagen\_0\_12m,  
 prediction = TRUE,  
 layout = "BMJ",  
 overall = FALSE,  
 subgroup = TRUE,  
 random = TRUE,  
 common = FALSE,  
 test.subgroup = TRUE,  
 print.stat = FALSE,  
 print.I2 = TRUE,  
 print.Q = TRUE,  
 print.pval.Q = TRUE,  
 print.tau2 = FALSE,  
 label.left = "Favors Control",  
 label.right = "Favors Intervention",  
 spacing = 1.2,  
 fontsize = 10,  
 digits = 2,  
 leftcols = c("studlab", "e\_n", "e\_reduc\_mean", "e\_reduc\_sd", "c\_n", "c\_reduc\_mean",  
 "c\_reduc\_sd"),  
 leftlabs = c("Study", "Intv. (n)", "Mean", "SD",   
 "Ctrl. (n)", "Mean", "SD"),  
 just.addcols = "center"  
 )  
  
  
# Add a footer for the test statistics of the effects after 0-3 months  
footer\_ter\_03\_red <- paste0(  
 "Subgroup effect | ",  
 paste0(  
 red\_ter\_subsplit\_012m$`0-3 months`,  
 ": z = ", round(red\_ter\_subsplit\_012m$`0-3 months`$z\_value, 2),  
 ", p = ", formatC(red\_ter\_subsplit\_012m$`0-3 months`$p\_value, digits = 2)  
 )  
)  
  
# Add a footer for the test statistics of the effects after 6 months  
footer\_ter\_6\_red <- paste0(  
 "Subgroup effect | ",  
 paste0(  
 red\_ter\_subsplit\_012m$`6 months`,  
 ": z = ", round(red\_ter\_subsplit\_012m$`6 months`$z\_value, 2),  
 ", p = ", formatC(red\_ter\_subsplit\_012m$`6 months`$p\_value, digits = 2)  
 )  
)  
  
# Add a footer for the test statistics of the effects after 12 months  
footer\_ter\_12\_red <- paste0(  
 "Subgroup effect | ",  
 paste0(  
 red\_ter\_subsplit\_012m$`12 months`,  
 ": z = ", round(red\_ter\_subsplit\_012m$`12 months`$z\_value, 2),  
 ", p = ", formatC(red\_ter\_subsplit\_012m$`12 months`$p\_value, digits = 2)  
 )  
)  
  
# Adjust the footer for the 0-3 month time point right below the subgroup  
grid.text(footer\_ter\_03\_red, x = 0.057, y = 0.52,  
 just = "left", gp = gpar(fontsize = 8))  
  
  
# Adjust the footer for the 6 month time point right below the subgroup  
grid.text(footer\_ter\_6\_red, x = 0.057, y = 0.25,  
 just = "left", gp = gpar(fontsize = 8))  
  
# Adjust the footer for the 12 month time point right below the subgroup  
grid.text(footer\_ter\_12\_red, x = 0.057, y = 0.05,  
 just = "left", gp = gpar(fontsize = 8))



1. Forest plot of short-, intermediate-, and long-term effects on alcohol reduction

### 4.2.5 Publication bias for intermediate effects on alcohol reduction

We only identified three studies for the intermediate time point, which makes it less meaningful to inspect for visual asymmetry, however, we proceed with creating the plot. First we will be changing the author names to fit the APA-format using the previously defined study labels.

# Assign the new study labels to the data frame  
red\_d\_full\_6m$study <- ifelse(red\_d\_full\_6m$study %in% names(study\_labels),  
 study\_labels[red\_d\_full\_6m$study],  
 red\_d\_full\_6m$study)

After defining the new labels, we can run a separate meta-analysis on the long-term time point.

# Run the meta-analysis stratified by superordinate domains and using the new labels  
red\_ter\_metagen\_6m <- metagen(TE = SMD,  
 seTE = SE,  
 studlab = study,  
 data = red\_d\_full\_6m,  
 sm = "SMD",  
 method.tau = "REML",  
 common = FALSE,  
 random = TRUE)

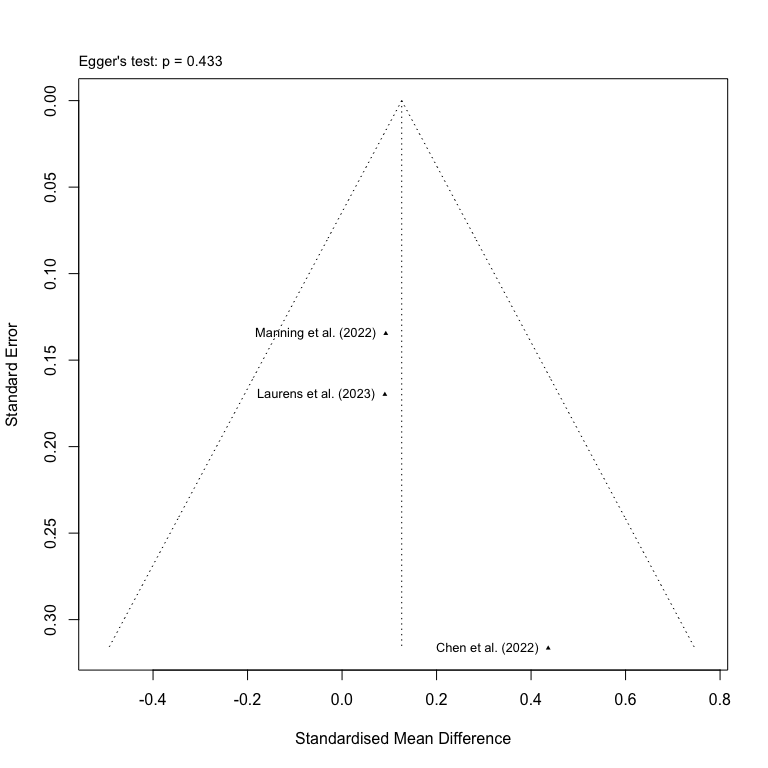
Though an Egger’s test will not have the required statistical power, any interpretation of the results should be highly cautioned. However, we proceed with conducting the test and adding it to the final plot.

# Conduct an Egger's test using the previously calculated treatment effects  
ca\_egger\_6m <- metabias(ca\_ter\_metagen\_6m, method.bias = "Egger", k.min = 2)  
  
# Print the results of the Egger's test  
print(ca\_egger\_6m)

## Linear regression test of funnel plot asymmetry  
##   
## Test result: t = -0.97, df = 2, p-value = 0.4334  
## Bias estimate: -0.7764 (SE = 0.7984)  
##   
## Details:  
## - multiplicative residual heterogeneity variance (tau^2 = 0.9887)  
## - predictor: standard error  
## - weight: inverse variance  
## - reference: Egger et al. (1997), BMJ

We can now create the funnel plot and add the result of the Egger’s test.

# Create a funnel plot for the 6 month time point  
funnel(red\_ter\_metagen\_6m,  
 xlab = "Standardised Mean Difference",  
 studlab = TRUE,  
 pch = 17,  
 cex = 0.5)  
  
# Add Egger's test statistic to the plot  
ca\_egger\_text\_6m <- paste0("Egger's test: p = ",   
 round(ca\_egger\_6m$p.value, 3))  
  
# Define the actual placement of the statistics on the funnel plot  
mtext(ca\_egger\_text\_6m, side=3, line=0.5, adj=0, cex=0.9, col="black")



1. Funnel plot of studies with intermediate effects on alcohol reduction

### 4.2.6 Publication bias for long-term effects on alcohol reduction

As we only identified one study for the long-term time point, a visual inspection for asymmetry will not be meaningful nor will there be sufficient power for an Egger’s test.

## 4.3 Intermediate and long-term effects on alcohol craving

The last part of our tertiary analyses will be conducted on alcohol craving, and we calculate the effect sizes for both the intermediate and long-term time point.

### 4.3.1 Calculation of effect sizes for intermediate continuous and categorical variables

We repeat the methodology used in section 4.2.1 and start with filtering the data for 6 months

# Filter 6 month data first and store it as a new subset  
crav\_d\_6m <- crav\_d %>% filter(time\_point == "6 months")

As we only have continuous data for alcohol craving, we only need to run the metacont()-function on our dataset to calculate the SMDs using Hedge’s *g* correction.

# Calculating the SMDs (Hedge's g) using metacont()  
crav\_metacont\_6m <- metacont(  
 n.e = e\_n,  
 mean.e = e\_crav\_mean,  
 sd.e = e\_crav\_sd,  
 n.c = c\_n,  
 mean.c = c\_crav\_mean,  
 sd.c = c\_crav\_sd,  
 studlab = study,  
 data = crav\_d\_6m,  
 sm = "SMD",  
 method.smd = "Hedges",  
 method.tau = "REML",  
 method.random.ci = "classic",  
 random = TRUE  
)

Now we can combine the calculated effect sizes and add them to the original data set without changing the original variable names.

# Combine the SMDs calculated with original variables and store it in a new data frame  
crav\_d\_full\_6m <- data.frame(  
 row\_id = crav\_d\_6m$row\_id,  
 study = crav\_metacont\_6m$studlab,  
 time\_point = crav\_d\_6m$time\_point,  
 super\_domain = crav\_d\_6m$super\_domain,  
 cog\_domain = crav\_d\_6m$cog\_domain,  
 total\_n = crav\_d\_6m$e\_n + crav\_d\_6m$c\_n, # Calculating total number of participants  
 c\_n = crav\_d\_6m$c\_n,  
 c\_crav\_mean = crav\_d\_6m$c\_crav\_mean,  
 c\_crav\_sd = crav\_d\_6m$c\_crav\_sd,  
 e\_n = crav\_d\_6m$e\_n,  
 e\_crav\_mean = crav\_d\_6m$e\_crav\_mean,  
 e\_crav\_sd = crav\_d\_6m$e\_crav\_sd,  
 # Adding calculated SMDs to the dataframe   
 SMD = crav\_metacont\_6m$TE,  
 SE = crav\_metacont\_6m$seTE,  
 LCI = crav\_metacont\_6m$lower,  
 UCI = crav\_metacont\_6m$upper,  
 # Adding the z- and p-values  
 `z-value` = crav\_metacont\_6m$TE / crav\_metacont\_6m$seTE,  
 `p-value` = 2 \* (1 - pnorm(abs(crav\_metacont\_6m$TE / crav\_metacont\_6m$seTE))),  
 # Adding a final note on how data was converted   
 Notes = "Calculated using metacont()",  
 check.names = FALSE # Removing the period as a space  
)

### 4.3.2 Calculation of effect sizes for intermediate continuous and categorical variables

We repeat the methodology used in the section above but this time filter the data by “12 months”

# Filter 6 month data first and store it as a new subset  
crav\_d\_12m <- crav\_d %>% filter(time\_point == "12 months")

As we only have continuous data for alcohol craving, we only need to run the metacont()-function on our dataset to calculate the SMDs using Hedge’s *g* correction.

# Calculating the SMDs (Hedge's g) using metacont()  
crav\_metacont\_12m <- metacont(  
 n.e = e\_n,  
 mean.e = e\_crav\_mean,  
 sd.e = e\_crav\_sd,  
 n.c = c\_n,  
 mean.c = c\_crav\_mean,  
 sd.c = c\_crav\_sd,  
 studlab = study,  
 data = crav\_d\_12m,  
 sm = "SMD",  
 method.smd = "Hedges",  
 method.tau = "REML",  
 method.random.ci = "classic",  
 random = TRUE  
)

Now we can combine the calculated effect sizes and add them to the original data set without changing the original variable names.

# Combine the SMDs calculated with original variables and store it in a new data frame  
crav\_d\_full\_12m <- data.frame(  
 row\_id = crav\_d\_12m$row\_id,  
 study = crav\_metacont\_12m$studlab,  
 time\_point = crav\_d\_12m$time\_point,  
 super\_domain = crav\_d\_12m$super\_domain,  
 cog\_domain = crav\_d\_12m$cog\_domain,  
 total\_n = crav\_d\_12m$e\_n + crav\_d\_12m$c\_n, # Calculating total number of participants  
 c\_n = crav\_d\_12m$c\_n,  
 c\_crav\_mean = crav\_d\_12m$c\_crav\_mean,  
 c\_crav\_sd = crav\_d\_12m$c\_crav\_sd,  
 e\_n = crav\_d\_12m$e\_n,  
 e\_crav\_mean = crav\_d\_12m$e\_crav\_mean,  
 e\_crav\_sd = crav\_d\_12m$e\_crav\_sd,  
 # Adding calculated SMDs to the dataframe   
 SMD = crav\_metacont\_12m$TE,  
 SE = crav\_metacont\_12m$seTE,  
 LCI = crav\_metacont\_12m$lower,  
 UCI = crav\_metacont\_12m$upper,  
 # Adding the z- and p-values  
 `z-value` = crav\_metacont\_12m$TE / crav\_metacont\_12m$seTE,  
 `p-value` = 2 \* (1 - pnorm(abs(crav\_metacont\_12m$TE / crav\_metacont\_12m$seTE))),  
 # Adding a final note on how data was converted   
 Notes = "Calculated using metacont()",  
 check.names = FALSE # Removing the period as a space  
)

### 4.3.3 Integrating the data for craving for all time points

Before we can integrate the data frames across all three time points, we need to assign the old variable name for “cognitive domain” to the data frame stored for 0-3 months.

# Assigning old variable names to the subgroups  
crav\_d\_full\_3m <- crav\_d\_full\_3m %>%  
 dplyr::rename(cog\_domain = Subgroup)

We can now integrate the two data frames with the data frame for 0-3 months using the the bind\_rows function.

# Combine the three data frames for craving on all time points and round all the values 5 decimals  
crav\_d\_full\_0\_12m <- bind\_rows(crav\_d\_full\_3m, crav\_d\_full\_6m, crav\_d\_full\_12m) %>% mutate(across(where(is.numeric), ~ round(.x, 5))) %>%  
 # Arrange the studies by their pre-defined row-numbers  
 arrange(row\_id)

To use the same author labels in APA-format across all studies and sub-studies, we need to apply the previously defined study labels to our integrated data set.

# Assign the new study labels to the combined data frame  
crav\_d\_full\_0\_12m$study <- ifelse(crav\_d\_full\_0\_12m$study %in% names(study\_labels),  
 study\_labels[crav\_d\_full\_0\_12m$study],  
 crav\_d\_full\_0\_12m$study)

### 4.3.4 Meta-analysis on intermediate and long-term craving

To make it clearer in the final forest plot, we assign new variable names to the stratification by time point, and sort them with the earliest time point on top.

# Assigning new variable names to the subgroups  
crav\_d\_full\_0\_12m <- crav\_d\_full\_0\_12m %>%  
 dplyr::rename("Time point" = time\_point) %>% # Renaming the time\_point variable  
 arrange(study) %>% # Arrange the studies alphabetically  
 arrange(desc(`Time point` == "0-3 months"), # Arrange then by Time Point  
 desc(`Time point` == "6 months"),  
 desc(`Time point` == "12 months")  
 )

Now with the newly defined subgroup labels, we can conduct the meta-analysis using metagen(), but this time set the parameter for subgroups to the label “Time point” and use the classic *z*-test for the random effects model.

# Run the meta-analysis stratified by superordinate domains and using the new labels  
crav\_ter\_metagen\_0\_12m <- metagen(TE = SMD,  
 seTE = SE,  
 studlab = study,  
 data = crav\_d\_full\_0\_12m,  
 subgroup = `Time point`,  
 sm = "OR",  
 method.tau = "REML",  
 common = FALSE,  
 random = TRUE,  
 method.random.ci = "classic")  
  
# Print the results of the meta-analysis  
print(crav\_ter\_metagen\_0\_12m)

## Number of studies: k = 12  
##   
## OR 95%-CI z p-value  
## Random effects model 0.7756 [0.5341; 1.1262] -1.34 0.1817  
##   
## Quantifying heterogeneity (with 95%-CIs):  
## tau^2 = 0.3718 [0.1567; 1.0950]; tau = 0.6097 [0.3958; 1.0464]  
## I^2 = 91.5% [87.1%; 94.4%]; H = 3.43 [2.78; 4.23]  
##   
## Test of heterogeneity:  
## Q d.f. p-value  
## 129.60 11 < 0.0001  
##   
## Results for subgroups (random effects model):  
## k OR 95%-CI tau^2 tau Q I^2  
## Time point = 0-3 months 9 0.9056 [0.6905; 1.1876] 0.1089 0.3300 26.35 69.6%  
## Time point = 6 months 2 0.4146 [0.0585; 2.9407] 1.9471 1.3954 38.89 97.4%  
## Time point = 12 months 1 0.9543 [0.7196; 1.2655] -- -- 0.00 --  
##   
## Test for subgroup differences (random effects model):  
## Q d.f. p-value  
## Between groups 0.71 2 0.7000  
##   
## Details of meta-analysis methods:  
## - Inverse variance method  
## - Restricted maximum-likelihood estimator for tau^2  
## - Q-Profile method for confidence interval of tau^2 and tau  
## - Calculation of I^2 based on Q

It is now possible to extract the test statistics for the individual time points, which can be used for the forest plot.

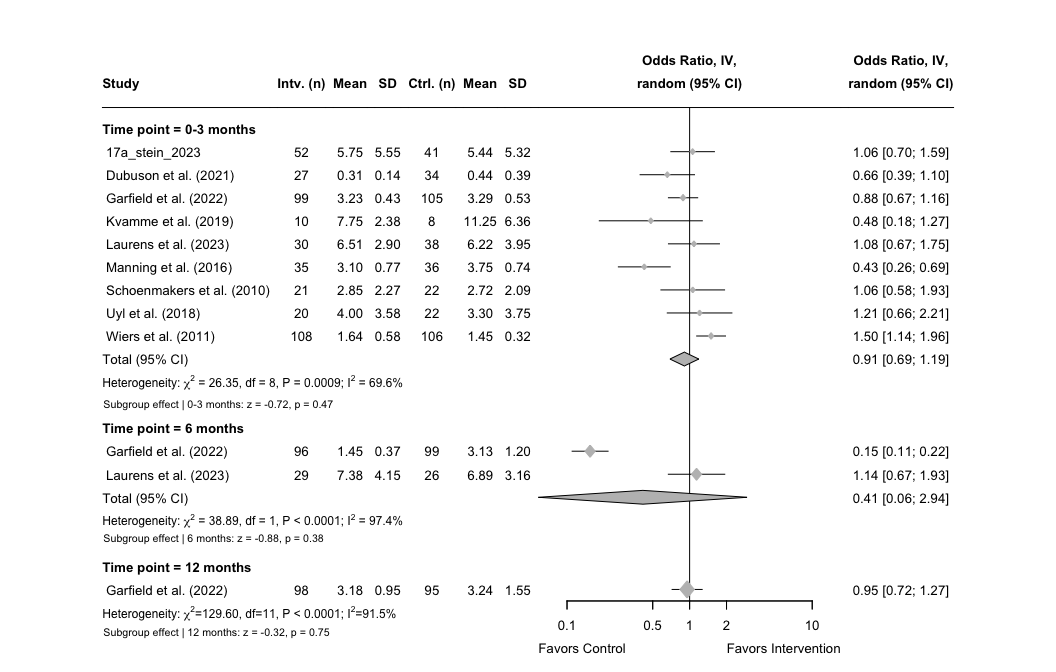
# Extract the test statistics for the subgroup analysis and create a data frame  
crav\_ter\_zp\_subgroup\_012m <- data.frame(  
 subgroup = crav\_ter\_metagen\_0\_12m$bylevs,  
 SMD = crav\_ter\_metagen\_0\_12m$TE.random.w,  
 SE = crav\_ter\_metagen\_0\_12m$seTE.random.w,  
 z\_value = crav\_ter\_metagen\_0\_12m$statistic.random.w,  
 p\_value = crav\_ter\_metagen\_0\_12m$pval.random.w  
)

For an easier way to add the individual effect sizes for each subgroup in the final forest plot, we can create a split of the data frame.

# Split the previously defined data frame by each subordinate domain  
crav\_ter\_subsplit\_012m <- split(crav\_ter\_zp\_subgroup\_012m, crav\_ter\_zp\_subgroup\_012m$subgroup)

Now that we have created a list for the test statistics for each of the time points, we can create the final forest plot. We set the overall effect to FALSE, as we do not want to pool the effects across reports of different time points in the same study.

# Now plot the forest plot again with new subgroup names  
forest(crav\_ter\_metagen\_0\_12m,  
 prediction = FALSE,  
 prediction.subgroup = FALSE,  
 layout = "BMJ",  
 subgroup = TRUE,  
 overall = FALSE,  
 random = TRUE,  
 common = FALSE,  
 test.subgroup = FALSE,  
 print.stat = FALSE,  
 print.I2 = TRUE,  
 print.Q = TRUE,  
 print.pval.Q = TRUE,  
 print.tau2 = FALSE,  
 label.left = "Favors Control",  
 label.right = "Favors Intervention",  
 spacing = 1.2,  
 fontsize = 10,  
 digits = 2,  
 leftcols = c("studlab", "e\_n", "e\_crav\_mean", "e\_crav\_sd", "c\_n", "c\_crav\_mean",  
 "c\_crav\_sd"),  
 leftlabs = c("Study", "Intv. (n)", "Mean", "SD",   
 "Ctrl. (n)", "Mean", "SD"),  
 just.addcols = "center"  
 )  
  
# Add a footer for the test statistics of the effects after 0-3 months  
footer\_ter\_03\_crav <- paste0(  
 "Subgroup effect | ",  
 paste0(  
 crav\_ter\_subsplit\_012m$`0-3 months`,  
 ": z = ", round(crav\_ter\_subsplit\_012m$`0-3 months`$z\_value, 2),  
 ", p = ", formatC(crav\_ter\_subsplit\_012m$`0-3 months`$p\_value, digits = 2)  
 )  
)  
  
# Add a footer for the test statistics of the effects after 6 months  
footer\_ter\_6\_crav <- paste0(  
 "Subgroup effect | ",  
 paste0(  
 crav\_ter\_subsplit\_012m$`6 months`,  
 ": z = ", round(crav\_ter\_subsplit\_012m$`6 months`$z\_value, 2),  
 ", p = ", formatC(crav\_ter\_subsplit\_012m$`6 months`$p\_value, digits = 2)  
 )  
)  
  
# Add a footer for the test statistics of the effects after 12 months  
footer\_ter\_12\_crav <- paste0(  
 "Subgroup effect | ",  
 paste0(  
 crav\_ter\_subsplit\_012m$`12 months`,  
 ": z = ", round(crav\_ter\_subsplit\_012m$`12 months`$z\_value, 2),  
 ", p = ", formatC(crav\_ter\_subsplit\_012m$`12 months`$p\_value, digits = 2)  
 )  
)  
  
# Adjust the footer for the 0-3 month time point right below the subgroup  
grid.text(footer\_ter\_03\_crav, x = 0.098, y = 0.4,  
 just = "left", gp = gpar(fontsize = 8))  
  
# Adjust the footer for the 6 month time point right below the subgroup  
grid.text(footer\_ter\_6\_crav, x = 0.098, y = 0.2,  
 just = "left", gp = gpar(fontsize = 8))  
  
# Adjust the footer for the 12 month time point right below the subgroup  
grid.text(footer\_ter\_12\_crav, x = 0.098, y = 0.06,  
 just = "left", gp = gpar(fontsize = 8))



1. Forest plot of short-, intermediate-, long-term effects on craving

### 4.3.5 Publication bias for intermediate effects on craving

As we only identified two studies for the intermediate time point, a visual inspection for asymmetry will not be meaningful nor will there be sufficient power for an Egger’s test.

### 4.3.6 Publication bias for long-term effects on craving

As we only identified one study for the long-term time point, a visual inspection for asymmetry will not be meaningful nor will there be sufficient power for an Egger’s test.

# 5. Exports of data and visual plots

In this final section, we extract the calculated effect sizes as a new Excel spreadsheet named “Calculated Effect Sizes”, which can be uploaded to a repository or as supplementary material for publication. Then we will be calculating the weights for the RoB-summary and exporting it as a new csv-file. Finally, we export the weighted RoB-summary plot and some of the forest plots of our meta-analyses in high resolution, which will be needed for the final manuscript.

## 5.1 Data export

For keeping our original terminology for our extracted data, we want to export our data set with the calculated effect sizes with an additional column containing the study ID, which refers to separate reports of the same study that is defined by a combination of a number and a letter (i.e., denoting a certain report for a specific time point). First, we define a new list reversing the previously defined study labels.

# Swap names and values  
study\_labels\_reversed <- setNames(names(study\_labels), study\_labels)

We can then add a column with the study ID to our full data frame for CA. However, we need to manually correct the APA author labels to the original study IDs.

# Add study ID using reversed labels  
ca\_d\_full\_0\_12m <- ca\_d\_full\_0\_12m %>%  
 mutate(study\_id = study\_labels\_reversed[study]) %>%  
   
# Correct specific study IDs based on row ID  
mutate(study\_id = case\_when(  
 row\_id == 1 ~ "3c\_dubuson\_2021",  
 row\_id == 10 ~ "16b\_schoenmakers\_2010",  
 row\_id == 13 ~ "3d\_dubuson\_2021",  
 row\_id == 14 ~ "5c\_manning\_2022",  
 row\_id == 15 ~ "9c\_kumar\_2019",  
 row\_id == 16 ~ "15b\_schenkel\_2024",  
 row\_id == 17 ~ "2b\_uyl\_2018",  
 row\_id == 18 ~ "3e\_dubuson\_2021",  
 row\_id == 20 ~ "5d\_manning\_2022",  
 row\_id == 21 ~ "12b\_peerenboom\_2022",  
 row\_id == 23 ~ "14b\_schenkel\_2023",  
 row\_id == 24 ~ "15c\_schenkel\_2024",  
 row\_id == 25 ~ "18b\_wiers\_2011",  
 TRUE ~ study\_id # Keep original study ID if no manual correction needed  
 )) %>%  
   
# Reorder columns: put study\_id immediately after row ID  
relocate(study\_id, .after = row\_id) %>%  
   
# Sort by row ID ascending  
arrange(row\_id)

This can be repeated for the full data frame for alcohol reduction, and manually change three of the APA author labels back to their original study IDs.

# Add study ID using reversed labels  
red\_d\_full\_0\_12m <- red\_d\_full\_0\_12m %>%  
 mutate(study\_id = study\_labels\_reversed[study]) %>%  
   
# Correct specific study IDs based on row ID  
mutate(study\_id = case\_when(  
 row\_id == 6 ~ "5c\_manning\_2022",  
 row\_id == 7 ~ "10c\_laurens\_2023",  
 row\_id == 8 ~ "5d\_manning\_2022",  
 TRUE ~ study\_id # Keep original study ID if no manual correction needed  
 )) %>%  
   
# Reorder columns: put study\_id immediately after row ID  
relocate(study\_id, .after = row\_id) %>%  
   
# Sort by row ID ascending  
arrange(row\_id)  
  
# Change the time point label back to original name  
names(red\_d\_full\_0\_12m)[names(red\_d\_full\_0\_12m) == "Time point"] <- "time\_point"

Finally, we can repeat the step for the full data frame for craving, and manually change the APA author labels back to their original study IDs.

# Add study ID using reversed labels  
crav\_d\_full\_0\_12m <- crav\_d\_full\_0\_12m %>%  
 mutate(study\_id = study\_labels\_reversed[study]) %>%  
   
# Correct specific study IDs based on row ID  
mutate(study\_id = case\_when(  
 row\_id == 10 ~ "5g\_garfield\_2022",  
 row\_id == 11 ~ "10c\_laurens\_2023",  
 row\_id == 12 ~ "5h\_garfield\_2022",  
 TRUE ~ study\_id # Keep original study ID if no manual correction needed  
 )) %>%  
   
# Reorder columns: put study\_id immediately after row ID  
relocate(study\_id, .after = row\_id) %>%  
   
# Sort by row ID ascending  
arrange(row\_id)  
  
# Change the time point label back to original name  
names(crav\_d\_full\_0\_12m)[names(crav\_d\_full\_0\_12m) == "Time point"] <- "time\_point"

The last step is to export all the data frames into one Excel sheet, but categorized by “Abstinence”, “Reduction” and “Craving”, which will all have their own sheet. We ensure that previously defined data frame for CA does not contain characters, which is achieved by converting the data columns.

Now we can turn to exporting the final data set for weighted RoB-summary. First, we will need to extract the unique studies used for our primary analyses.

# Define study IDs for CA  
ca\_rob <- c(  
 "6\_grieder\_2022", "8b\_kvamme\_2019", "16b\_schoenmakers\_2010", "2b\_uyl\_2018",  
 "9b\_kumar\_2019", "3c\_dubuson\_2021", "1\_chen\_2022", "17b\_stein\_2023",  
 "18b\_wiers\_2011", "5b\_manning\_2022", "12a\_peerenboom\_2022", "14a\_schenkel\_2023",  
 "4\_eberl\_2013", "13a\_rinck\_2018", "15a\_schenkel\_2024", "11\_manning\_2016"  
)  
  
# Define study IDs for alcohol reduction  
red\_rob <- c(  
 "7c\_garfield\_2024", "10b\_laurens\_2023"  
)  
  
# Filter and select only needed columns in the data frame for CA  
ca\_d\_rob <- ca\_d\_full\_0\_12m %>%  
 filter(study\_id %in% ca\_rob) %>%  
 select(study\_id, study, SE) # <- only these columns  
  
# Filter and select only needed columns in the data frame for alcohol reduction  
red\_d\_rob <- red\_d\_full\_0\_12m %>%  
 filter(study\_id %in% red\_rob) %>%  
 select(study\_id, study, SE)  
  
# Combine the two sets  
rob\_d\_se <- bind\_rows(ca\_d\_rob, red\_d\_rob)

With a new data frame only including the outcomes extracted and rated from our unique studies, we can calculate the absolute and relative weights using the SE.

# Calculate weights  
rob\_d\_weighted <- rob\_d\_se %>%  
 mutate(weight = 1 / (SE^2)) %>%  
 mutate(weight\_percent = 100 \* weight / sum(weight))

The relative weights can now be added to the final RoB-summary data and exported as a new csv-file named “Weighted RoB Data”. First, we need to ensure that the column named “Study” matches the new data frame, and we will also need to filter out two reports that are not unique studies Manning et al. (2021).

# Rename 'study' to 'Study' in weighted\_rob\_d  
rob\_d\_weighted <- rob\_d\_weighted %>%  
 rename(Study = study)  
  
# Merge weights (adjust column names if necessary)  
rob\_d\_weighted <- rob\_d %>%  
 left\_join(rob\_d\_weighted %>% select(Study, weight\_percent),  
 by = "Study")  
  
# Remove double-reported studies  
rob\_d\_weighted <- rob\_d\_weighted %>%  
 filter(!Study %in% c("Garfield et al. (2022)\*", "Manning et al. (2021)\*"))  
  
# Save new CSV as "Weighted RoB Data" in the working directory  
write.csv(rob\_d\_weighted, "Weighted RoB Data.csv", row.names = FALSE)

Now we want to display the percentages, so that these can be reported in the manuscript.

# Reshape to long format (excluding weight column)  
rob\_pct <- rob\_d\_weighted %>%  
 pivot\_longer(cols = D1:Overall, names\_to = "Domain", values\_to = "Judgment")  
  
# Label domains and enforce correct domain order using factor  
rob\_labels <- c(  
 D1 = "Randomization process",  
 D2 = "Deviations from intended interventions",  
 D3 = "Missing outcome data",  
 D4 = "Measurement of the outcome",  
 D5 = "Selection of the reported result",  
 Overall = "Overall risk of bias"  
)  
  
# Use the label values to define order  
domain\_order <- rob\_labels %>% unname()  
rob\_pct <- rob\_pct %>%  
 mutate(  
 Domain = recode(Domain, !!!rob\_labels),  
 Domain = factor(Domain, levels = domain\_order),  
 Judgment = factor(Judgment, levels = c("Low", "Some concerns", "High")) # <-- key line  
 )  
  
# Count and calculate percentage  
rob\_pct\_summary <- rob\_pct %>%  
 group\_by(Domain, Judgment) %>%  
 summarise(n = n(), .groups = "drop") %>%  
 group\_by(Domain) %>%  
 mutate(Percent = round(n / sum(n) \* 100, 1))  
  
# Wide format summary table  
rob\_pct\_summary <- rob\_pct\_summary %>%  
 select(Domain, Judgment, Percent) %>%  
 pivot\_wider(names\_from = Judgment, values\_from = Percent, values\_fill = 0)  
  
# Print the ordered summary table  
print(rob\_pct\_summary)

## # A tibble: 6 × 4  
## # Groups: Domain [6]  
## Domain Low `Some concerns` High  
## <fct> <dbl> <dbl> <dbl>  
## 1 Randomization process 72.2 27.8 0   
## 2 Deviations from intended interventions 77.8 22.2 0   
## 3 Missing outcome data 44.4 33.3 22.2  
## 4 Measurement of the outcome 77.8 11.1 11.1  
## 5 Selection of the reported result 61.1 38.9 0   
## 6 Overall risk of bias 22.2 44.4 33.3

## 5.2 Figure export

In this section we export some of the figures displayed in this report. As the final manuscript will not include all of the figures displayed in this report, we only want to create high resolution exports for the files used in the manuscript. The remaining plots and figures will be added to the supplementary material in a word-file using the images created by knitr and the corresponding knitted word-document. We will export four figures in total and they follow the naming convention “fig\_x\_”, and as a flow-chart is included as the first figure in the manuscript, the figures will start with “fig\_2\_”.

We first export the summary of the RoB-assessment across all domains, but this time with the weighted summary data. The figure will be exported as figure 2.

We then export our forest plots, and as most of our identified studies included CA, we will only export the forest plots for our primary, secondary, and tertiary analyses for CA. The plots for alcohol reduction and craving will be included in the supplementary material.

The first forest plot for our primary analysis on short-term effects on CA, will be exported as figure 3.

# Export the forest plot for CA for 0-3 months  
tiff("fig\_3\_ca\_primary\_forest\_plot.tiff", width = 12, height = 7, units = "in", res = 600)  
forest(ca\_metagen\_3m,  
 prediction = TRUE,  
 prediction.subgroup = TRUE,  
 layout = "BMJ",  
 subgroup = TRUE,  
 overall = TRUE,  
 random = TRUE,  
 common = FALSE,  
 test.subgroup = FALSE,  
 print.stat = FALSE,  
 print.I2 = TRUE,  
 print.Q = TRUE,  
 print.pval.Q = TRUE,  
 print.tau2 = FALSE,  
 label.left = "Favors Control",  
 label.right = "Favors Intervention",  
 spacing = 1.2,  
 fontsize = 10,  
 digits = 2,  
 leftcols = c("studlab", "e\_abs", "e\_n", "c\_abs", "c\_n"),  
 leftlabs = c("Study", "Intv. (events)", "Total (n)",   
 "Ctrl. (events)", "Total (n)"),  
 just.addcols = "center"  
 )  
# Add a footer for the test statistics of the effects for explicit cognition  
footer\_exp\_ca\_3m <- paste0(  
 "Subgroup effect | ",  
 paste0(  
 ca\_zp\_subsplit\_3m$`Explicit cognition`,  
 ": z = ", round(ca\_zp\_subsplit\_3m$`Explicit cognition`$z\_value, 2),  
 ", p = ", formatC(ca\_zp\_subsplit\_3m$`Explicit cognition`$p\_value, digits = 2)  
 )  
)  
# Add a footer for the test statistics of the effects for implicit cognition  
footer\_imp\_ca\_3m <- paste0(  
 "Subgroup effect | ",  
 paste0(  
 ca\_zp\_subsplit\_3m$`Implicit cognition`,  
 ": z = ", round(ca\_zp\_subsplit\_3m$`Implicit cognition`$z\_value, 2),  
 ", p = ", formatC(ca\_zp\_subsplit\_3m$`Implicit cognition`$p\_value, digits = 2)  
 )  
)  
# Add a footer combining the test statistics for the overall effect  
footer\_over\_ca\_3m <- paste0(  
 "Overall effect: ",  
 paste0(  
 "z = ", round(ca\_zp\_overall\_3m$z\_value, 2),  
 ", p = ", formatC(ca\_zp\_overall\_3m$p\_value, digits = 2),  
 collapse = "; "  
 )  
)  
# Adjust the footer for the subgroup effects after explicit subgroup  
grid.text(footer\_exp\_ca\_3m, x = 0.092, y = 0.72,  
 just = "left", gp = gpar(fontsize = 8))  
# Adjust the footer for the subgroup effects after implicit subgroup  
grid.text(footer\_imp\_ca\_3m, x = 0.092, y = 0.21,  
 just = "left", gp = gpar(fontsize = 8))  
# Adjust the footer for the overall effect to the bottom left  
grid.text(footer\_over\_ca\_3m, x = 0.092, y = 0.045,  
 just = "left", gp = gpar(fontsize = 9))  
invisible(dev.off())

The second forest plot for our secondary analysis on short-term effects on CA, will be exported as figure 4.

# Export the forest plot for secondary analysis for CA (0-3 months)  
tiff("fig\_4\_ca\_secondary\_forest\_plot.tiff", width = 12, height = 9, units = "in", res = 600)  
forest(ca\_sec\_metagen\_3m,  
 prediction = TRUE,  
 prediction.subgroup = TRUE,  
 layout = "BMJ",  
 subgroup = TRUE,  
 overall = TRUE,  
 random = TRUE,  
 common = FALSE,  
 test.subgroup = FALSE,  
 print.stat = FALSE,  
 print.I2 = TRUE,  
 print.Q = TRUE,  
 print.pval.Q = TRUE,  
 print.tau2 = FALSE,  
 label.left = "Favors Control",  
 label.right = "Favors Intervention",  
 spacing = 1.2,  
 fontsize = 10,  
 digits = 2,  
 leftcols = c("studlab", "e\_abs", "e\_n", "c\_abs", "c\_n"),  
 leftlabs = c("Study", "Intv. (events)", "Total (n)",   
 "Ctrl. (events)", "Total (n)"),  
 just.addcols = "center"  
 )  
# Add a footer for the test statistics of the effects for executive functions  
footer\_sec\_exft\_ca\_3m <- paste0(  
 "Subgroup effect | ",  
 paste0(  
 ca\_sec\_subsplit\_3m$`Executive functions`,  
 ": z = ", round(ca\_sec\_subsplit\_3m$`Executive functions`$z\_value, 2),  
 ", p = ", formatC(ca\_sec\_subsplit\_3m$`Executive functions`$p\_value, digits = 2)  
 )  
)  
# Add a footer for the test statistics of the effects for approach bias  
footer\_sec\_apbm\_ca\_3m <- paste0(  
 "Subgroup effect | ",  
 paste0(  
 ca\_sec\_subsplit\_3m$`Approach bias`,  
 ": z = ", round(ca\_sec\_subsplit\_3m$`Approach bias`$z\_value, 2),  
 ", p = ", formatC(ca\_sec\_subsplit\_3m$`Approach bias`$p\_value, digits = 2)  
 )  
)  
# Add a footer for the test statistics of the effects for attentional bias  
footer\_sec\_atbm\_ca\_3m <- paste0(  
 "Subgroup effect | ",  
 paste0(  
 ca\_sec\_subsplit\_3m$`Attentional bias`,  
 ": z = ", round(ca\_sec\_subsplit\_3m$`Attentional bias`$z\_value, 2),  
 ", p = ", formatC(ca\_sec\_subsplit\_3m$`Attentional bias`$p\_value, digits = 2)  
 )  
)  
# Add a footer for the test statistics of the effects for inhibitory bias  
footer\_sec\_itbm\_ca\_3m <- paste0(  
 "Subgroup effect | ",  
 paste0(  
 ca\_sec\_subsplit\_3m$`Inhibitory bias`,  
 ": z = ", round(ca\_sec\_subsplit\_3m$`Inhibitory bias`$z\_value, 2),  
 ", p = ", formatC(ca\_sec\_subsplit\_3m$`Inhibitory bias`$p\_value, digits = 2)  
 )  
)  
# Add a footer combining the test statistics for the overall effect  
footer\_sec\_over\_ca\_3m <- paste0(  
 "Overall effect: ",  
 paste0(  
 "z = ", round(ca\_sec\_zp\_overall\_3m$z\_value, 2),  
 ", p = ", formatC(ca\_sec\_zp\_overall\_3m$p\_value, digits = 2),  
 collapse = "; "  
 )  
)  
# Adjust the footer for the executive functions right below the subgroup  
grid.text(footer\_sec\_exft\_ca\_3m, x = 0.091, y = 0.805,  
 just = "left", gp = gpar(fontsize = 8))  
# Adjust the footer for the approach bias right below the subgroup  
grid.text(footer\_sec\_apbm\_ca\_3m, x = 0.091, y = 0.57,  
 just = "left", gp = gpar(fontsize = 8))  
# Adjust the footer for the attentional bias right below the subgroup  
grid.text(footer\_sec\_atbm\_ca\_3m, x = 0.091, y = 0.378,  
 just = "left", gp = gpar(fontsize = 8))  
# Adjust the footer for the inhibitory bias right below the subgroup  
grid.text(footer\_sec\_itbm\_ca\_3m, x = 0.091, y = 0.145,  
 just = "left", gp = gpar(fontsize = 8))  
# Adjust the footer for the overall effect to the bottom left  
grid.text(footer\_sec\_over\_ca\_3m, x = 0.090, y = 0.025,  
 just = "left", gp = gpar(fontsize = 9))  
invisible(dev.off())

The final forest plot for our tertiary analysis on short-, intermediate-, and long-term effects on CA, will be exported as figure 5.

# Export the forest plot for tertiary analysis for CA (all time points)  
tiff("fig\_5\_ca\_tertiary\_forest\_plot.tiff", width = 12, height = 12, units = "in", res = 600)  
forest(ca\_ter\_metagen\_0\_12m,  
 prediction = TRUE,  
 prediction.subgroup = TRUE,  
 layout = "BMJ",  
 subgroup = TRUE,  
 overall = FALSE,  
 random = TRUE,  
 common = FALSE,  
 test.subgroup = FALSE,  
 print.stat = FALSE,  
 print.I2 = TRUE,  
 print.Q = TRUE,  
 print.pval.Q = TRUE,  
 print.tau2 = FALSE,  
 label.left = "Favors Control",  
 label.right = "Favors Intervention",  
 spacing = 1.2,  
 fontsize = 10,  
 digits = 2,  
 leftcols = c("studlab", "e\_abs", "e\_n", "c\_abs", "c\_n"),  
 leftlabs = c("Study", "Intv. (events)", "Total (n)",   
 "Ctrl. (events)", "Total (n)"),  
 just.addcols = "center"  
 )  
# Add a footer for the test statistics of the effects after 0-3 months  
footer\_ter\_03\_ca <- paste0(  
 "Subgroup effect | ",  
 paste0(  
 ca\_ter\_subsplit\_012m$`0-3 months`,  
 ": z = ", round(ca\_ter\_subsplit\_012m$`0-3 months`$z\_value, 2),  
 ", p = ", formatC(ca\_ter\_subsplit\_012m$`0-3 months`$p\_value, digits = 2)  
 )  
)  
# Add a footer for the test statistics of the effects after 6 months  
footer\_ter\_6\_ca <- paste0(  
 "Subgroup effect | ",  
 paste0(  
 ca\_ter\_subsplit\_012m$`6 months`,  
 ": z = ", round(ca\_ter\_subsplit\_012m$`6 months`$z\_value, 2),  
 ", p = ", formatC(ca\_ter\_subsplit\_012m$`6 months`$p\_value, digits = 2)  
 )  
)  
# Add a footer for the test statistics of the effects after 12 months  
footer\_ter\_12\_ca <- paste0(  
 "Subgroup effect | ",  
 paste0(  
 ca\_ter\_subsplit\_012m$`12 months`,  
 ": z = ", round(ca\_ter\_subsplit\_012m$`12 months`$z\_value, 2),  
 ", p = ", formatC(ca\_ter\_subsplit\_012m$`12 months`$p\_value, digits = 2)  
 )  
)  
# Adjust the footer for the 0-3 month time point right below the subgroup  
grid.text(footer\_ter\_03\_ca, x = 0.114, y = 0.56,  
 just = "left", gp = gpar(fontsize = 8))  
# Adjust the footer for the 6 month time point right below the subgroup  
grid.text(footer\_ter\_6\_ca, x = 0.114, y = 0.3825,  
 just = "left", gp = gpar(fontsize = 8))  
# Cover up unwanted group heterogeneity statistics  
grid.rect(  
 x = 0.1,  
 y = 0.1,  
 width = 0.6,  
 height = 0.02,  
 gp = gpar(fill = "white", col = NA)  
)  
# Adjust the footer for the 12 month time point right below the subgroup  
grid.text(footer\_ter\_12\_ca, x = 0.114, y = 0.1,  
 just = "left", gp = gpar(fontsize = 8))  
invisible(dev.off())

# 6. R packages used

Some of the auxiliary packages used in this analysis report can be outputted using the package grateful by Rodriguez-Sanchez [aut et al. (2025). The version number and the corresponding reference of the package are listed next to the package name in the package below.

# Add summary of the packages used except the meta-package  
cite\_packages(output = "table", out.dir = ".")

## Package Version Citation  
## 1 base 4.5.1 @base  
## 2 devtools 2.4.5 @devtools  
## 3 knitr 1.50 @knitr2014; @knitr2015; @knitr2025  
## 4 renv 1.1.4 @renv  
## 5 rmarkdown 2.29 @rmarkdown2018; @rmarkdown2020; @rmarkdown2024  
## 6 robvis 0.3.0.900 @robvis  
## 7 tidyverse 2.0.0 @tidyverse

# 7. References

Balduzzi, Sara, Gerta Rücker, and Guido Schwarzer. 2019. “How to perform a meta-analysis with R: a practical tutorial.” *Evidence-Based Mental Health* 22 (4): 153–60. <https://doi.org/10.1136/ebmental-2019-300117>.

Chinn, Susan. 2000. “A Simple Method for Converting an Odds Ratio to Effect Size for Use in Meta-Analysis.” *Statistics in Medicine* 19 (22): 3127–31. <https://doi.org/10.1002/1097-0258(20001130)19:22<3127::AID-SIM784>3.0.CO;2-M>.

Garfield, Joshua B.B., Lara R. Piccoli, Danielle Whelan, Petra K. Staiger, John Reynolds, Hugh Piercy, Dan I. Lubman, Antonio Verdejo-Garcia, and Victoria Manning. 2022. “The Effect of Approach Bias Modification During Alcohol Withdrawal Treatment on Craving, and Its Relationship to Post-Treatment Alcohol Use in a Randomised Controlled Trial.” *Drug and Alcohol Dependence* 239 (October): 109621. <https://doi.org/10.1016/j.drugalcdep.2022.109621>.

Grieder, Matthias, Leila M. Soravia, Raphaela M. Tschuemperlin, Hallie M. Batschelet, Andrea Federspiel, Simon Schwab, Franz Moggi, and Maria Stein. 2022. “Right Inferior Frontal Activation During Alcohol-Specific Inhibition Increases with Craving and Predicts Drinking Outcome in Alcohol Use Disorder.” <https://doi.org/10.31234/osf.io/h6t7f>.

Manning, Victoria, Hugh Piercy, Joshua Benjamin Bernard Garfield, Stuart Gregory Clark, Mah Noor Andrabi, and Dan Ian Lubman. 2021. “A Personalized Approach Bias Modification Smartphone App (‘SWiPE’) to Reduce Alcohol Use: Open-Label Feasibility, Acceptability, and Preliminary Effectiveness Study.” *JMIR mHealth and uHealth* 9 (12): e31353. <https://doi.org/10.2196/31353>.

Rodriguez-Sanchez [aut, Francisco, Connor P. Jackson, Shaurita D. Hutchins, and James M. Clawson. 2025. *Grateful: Facilitate Citation of R Packages*. <https://cran.r-project.org/web/packages/grateful/index.html>.

Stein, Maria, Leila M. Soravia, Raphaela M. Tschuemperlin, Hallie M. Batschelet, Joshua Jaeger, Susanne Roesner, Anne Keller, Juan Martin Gomez Penedo, Reinout W. Wiers, and Franz Moggi. 2023. “Alcohol‐specific Inhibition Training in Patients with Alcohol Use Disorder: A Multi‐centre, Double‐blind Randomized Clinical Trial Examining Drinking Outcome and Working Mechanisms.” *Addiction* 118 (4): 646–57. <https://doi.org/10.1111/add.16104>.

Sterne, Jonathan A C, Jelena Savović, Matthew J Page, Roy G Elbers, Natalie S Blencowe, Isabelle Boutron, Christopher J Cates, et al. 2019. “RoB 2: a revised tool for assessing risk of bias in randomised trials.” *British Medical Journal* 366 (14898): 1–8. <https://doi.org/10.1136/bmj.l4898>.