

Supplemental Material 3

Meta-Analysis of Digital Neuropsychological Interventions

Contents

1. Introduction	1
1.1 Loading and installing the packages	2
1.3 Importing risk of bias data	3
1.4 Displaying the risk of bias	3
2. Main analyses on short-term outcomes	9
2.1 Short-term abstinence	9
2.2 Short-term alcohol reduction	24
2.3 Short-term alcohol craving	29
3. Main analyses on intermediate-term outcomes	35
3.1 Intermediate-term abstinence	35
3.2 Intermediate-term alcohol reduction	43
3.3 Intermediate-term craving	48
4. Main analyses on long-term outcomes	50
4.1 Long-term abstinence	50
4.2 Long-term effects on alcohol reduction	57
4.3 Long-term effects on craving	58
5. Exploratory intervention-specific analyses	59
5.1 Short-term effects on abstinence by intervention	59
5.2 Short-term effects on alcohol reduction by intervention	63
5.3 Short-term effects on alcohol craving by intervention	64
5.4 Intermediate-term effects on abstinence by intervention	66
5.5 Intermediate-term effects on alcohol reduction by intervention	69
5.6 Intermediate-term effects on craving by intervention	71
5.7 Long-term effects on abstinence by intervention	71
5.8 Long-term effects on alcohol reduction by intervention	74
5.9 Long-term effects on craving by intervention	75
6. Exports of data and visual plots	75
6.1 Data export	75
6.2 Figure export	83
7. R packages used	97
8. References	98

1. Introduction

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

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

The packages needed for this report are `rmarkdown`, `knitr`, `here`, `osfr`, `devtools`, `robvis`, `readxl`, `openxlsx`, `dplyr`, `tidyr`, `tidyverse`, `meta`, `ggplot2`, `grid`, `gtable`, `cowplot`, and `grateful`. The `rmarkdown` and `knitr` packages are used for knitting the R markdown file to PDF or word, the `here` package is used to build file paths relative to the project root (so the report works regardless of the working directory), and `osfr` is used to retrieve the project data from the OSF repository. The `devtools` package is used to install the developer version of `robvis`, which is required as this version displays the study labels horizontally; `robvis` itself is used to create the traffic light plot and the summary of the risk of bias ratings. The `readxl` and `openxlsx` packages are used for reading and writing the Excel data sets, whereas the `dplyr`, `tidyr`, and `tidyverse` packages are 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 `ggplot2`, `grid`, and `gtable` packages are used to build and fine-tune the plots (for example, adding statistics and frame lines that are not standard for the `meta` or `robvis` packages), and the `cowplot` package is used to combine the individual funnel plots into a single figure. 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-npi-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.

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.

```
# Local directory for storing the downloaded data, anchored to the project root
# with here() so the path resolves whether the main file is knitted or a child
# file is run on its own. Keeping the data on disk means it is only fetched from
# OSF once, which greatly speeds up re-rendering.
data_dir <- here::here("data")
if (!dir.exists(data_dir)) dir.create(data_dir)

# Directories for the generated outputs (created up front so the figure and data
# exports later in the report always have somewhere to write to)
dir.create(here::here("outputs", "figures"), showWarnings = FALSE, recursive = TRUE)
dir.create(here::here("outputs", "data"),      showWarnings = FALSE, recursive = TRUE)
dir.create(here::here("assets"),               showWarnings = FALSE, recursive = TRUE)
dir.create(here::here("reports"),              showWarnings = FALSE, recursive = TRUE)

# Expected local file paths for the two data sets
rob_local      <- file.path(data_dir, "Risk of Bias Data.csv")
clinical_local <- file.path(data_dir, "Extracted Clinical Outcome Data.xlsx")

# Only contact OSF if the data has not already been stored locally
if (!file.exists(rob_local) || !file.exists(clinical_local)) {

  # Get the OSF project node
  osf_proj <- osf_retrieve_node("9xtsq")
```

```

# List all folders inside the project
osf_folders <- osf_ls_files(osf_proj, type = "folder")

# Retrieve the "Risk of bias data" and "Clinical outcome data" folders
osf_rob_folder <- osf_retrieve_file(
  osf_folders %>% filter(name == "Risk of bias data") %>% pull(id))
osf_clinical_folder <- osf_retrieve_file(
  osf_folders %>% filter(name == "Clinical outcome data") %>% pull(id))

# Download the desired files into the local data directory
osf_ls_files(osf_rob_folder) %>%
  filter(name == "Risk of Bias Data.csv") %>%
  osf_download(path = data_dir, conflicts = "overwrite")

osf_ls_files(osf_clinical_folder) %>%
  filter(name == "Extracted Clinical Outcome Data.xlsx") %>%
  osf_download(path = data_dir, conflicts = "overwrite")
}

# Expose the local paths through the same interface the rest of the report uses,
# so no downstream code needs to change
osf_rob_files <- list(local_path = rob_local)
osf_clinical_files <- list(local_path = clinical_local)

```

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
traffic_plot <- rob_traffic_light(rob_d, tool = "ROB2", psize = 7)

# Remove alternating grey row stripes
traffic_plot$layers[[4]]$aes_params$fill <- "#ffffff"
traffic_plot$layers[[4]]$aes_params$colour <- "#ffffff"

# Slight nudge right for symbols (+, -, X) to centre within circles
traffic_plot$layers[[2]]$position <- position_nudge(x = 0.0075)
traffic_plot$layers[[7]]$position <- position_nudge(x = 0.0072, y = 0.01)

# position_nudge only affects the plot, not the legend key, so override the
# key drawing for the symbol layer to shift the symbol right to match the peas

```

```

traffic_plot$layers[[2]]$geom <- ggproto("GeomPointNudge", GeomPoint,
  draw_key = function(data, params, size) {
    key <- draw_key_point(data, params, size)
    key$x <- key$x + unit(0.4, "pt")
    key
  })

# Restore black circle outline on the peas in the plot
traffic_plot$layers[[6]]$aes_params$colour <- "black"

# The legend key is built from layer 1 (the disc). Give it a fill-mapped,
# black-ringed shape 21 so the key shows a colored pea WITH a black outline.
# In the plot this layer sits under the larger size-7 disc (layer 5), so this
# affects only the legend key, not the plotted peas.
traffic_plot$layers[[1]]$mapping <- aes(fill = judgement)
traffic_plot$layers[[1]]$aes_params$shape <- 21
traffic_plot$layers[[1]]$aes_params$colour <- "black"
traffic_plot$layers[[1]]$aes_params$stroke <- 0.5

# Rename Overall to 0 in all data sources
traffic_plot$data$domain[traffic_plot$data$domain == "Overall"] <- "0A"
for (i in seq_along(traffic_plot$layers)) {
  d <- traffic_plot$layers[[i]]$data
  if (!is.null(d) && nrow(d) > 0 && "domain" %in% names(d)) {
    traffic_plot$layers[[i]]$data$domain[d$domain == "Overall"] <- "0A"
  }
}
rm(i, d) # drop loop temporaries so they don't linger in the environment
traffic_plot <- traffic_plot +
  facet_grid(Study ~ factor(domain, levels = c("D1", "D2", "D3", "D4", "D5", "0A")),
    switch = "y", space = "free")

# Linewidth for plot lines, and a thinner one for the judgement box
legend_lw <- 0.4
legend_box_lw <- 0.25

# Build the styled plot
traffic_plot <- traffic_plot +
  scale_y_continuous(limits = c(0.5, 1.5), labels = NULL, breaks = NULL,
    name = NULL, position = "left",
    # Symmetric expansion centres each pea in its cell so the top
    # row does not touch the header line (row spacing is unchanged)
    expand = expansion(add = c(0.15, 0.15))) +
  scale_x_discrete(position = "top", name = NULL) +
  labs(caption = NULL) +
  scale_colour_manual(
    values = c(h = "#BF0000", s = "#E2DF07", l = "#02C100",
      n = "#4EA1F7", x = "#cccccc"),
    breaks = c("h", "s", "l"),
    labels = c("High risk", "Some concerns", "Low risk"),
    name = "Judgement"
  ) +
  scale_shape_manual(

```

```

values = c(l = 43, s = 45, h = 120, n = 63, x = 32),
breaks = c("h", "s", "l"),
labels = c("High risk", "Some concerns", "Low risk"),
name = "Judgement"
) +
# Fill scale drives the coloured disc in the legend key (see layer 1 above)
scale_fill_manual(
  values = c(h = "#BF0000", s = "#E2DF07", l = "#02C100",
             n = "#4EA1F7", x = "#cccccc"),
  breaks = c("h", "s", "l"),
  labels = c("High risk", "Some concerns", "Low risk"),
  name = "Judgement"
) +
guides(
  # Hide the raw colour legend; the fill guide draws the ringed coloured pea
  # and the shape guide overlays the +/-x symbol on top of it.
  colour = "none",
  fill = guide_legend(override.aes = list(size = 7), direction = "vertical"),
  shape = guide_legend(override.aes = list(size = 4.5), direction = "vertical")
) +
theme(
  strip.background.y = element_rect(fill = "white", colour = NA),
  strip.background.x = element_rect(fill = "white", colour = NA),
  strip.text.x = element_text(face = "bold", size = 10,
                              margin = margin(t = 2, b = 12, l = 0, r = 0)),
  strip.text.y.left = element_text(angle = 0, size = 10, hjust = 0),
  panel.border = element_blank(),
  panel.spacing.x = unit(-3, "pt"),
  panel.spacing.y = unit(0, "pt"),
  legend.position = "bottom",
  legend.direction = "vertical",
  legend.justification = "right",
  legend.margin = margin(t = 4, r = 6, b = 6, l = 6),
  legend.box.margin = margin(t = 14, r = 0, b = 4, l = 0),
  legend.background = element_rect(fill = "white", colour = "black",
                                    linetype = "solid", linewidth = legend_box_lw),
  legend.key = element_rect(fill = "white"),
  legend.key.size = unit(1.0, "lines"),
  legend.text = element_text(size = 10),
  legend.title = element_text(size = 10, face = "plain")
)

# Add header line, bottom line, and vertical separator via gtable.
# Building the plot here is where ggplot2 raises a few advisory warnings
# ("Duplicated override.aes", "All aesthetics have length 1...") caused by the
# layer surgery above (re-mapping layer 1 to draw the ringed legend key). The
# resulting plot is visually correct (verified), so these are expected and
# suppressed at their source rather than hidden with a blanket chunk option.
traffic_gt <- suppressWarnings(ggplot_gtable(ggplot_build(traffic_plot)))

panel_layout <- traffic_gt$layout[grepl("^panel-", traffic_gt$layout$name), ]
strip_l <- traffic_gt$layout[grepl("^strip-l", traffic_gt$layout$name), ]
strip_t <- traffic_gt$layout[grepl("^strip-t", traffic_gt$layout$name), ]

```

```

top_row <- min(strip_t$t)
header_bottom <- max(strip_t$b)
bottom_row <- max(panel_layout$b)
left_col <- min(strip_l$l)
right_col <- max(panel_layout$r)
sep_col <- min(panel_layout$l)

# Match legend box line thickness (convert ggplot2 mm to grid pt)
line_col <- "black"
line_lwd <- legend_box_lw * ggplot2::.pt

# Nudge the horizontal frame lines away from the peas by a fixed amount
gap_top <- unit(5, "pt") # header line raised (also moves it toward the labels)
gap_bot <- unit(8, "pt") # bottom line lowered (away from the last pea row)

# Header bottom line
traffic_gt <- gtable::gtable_add_grob(traffic_gt,
  grid::segmentsGrob(x0 = 0, x1 = 1, y0 = gap_top, y1 = gap_top,
    gp = grid::gpar(col = line_col, lwd = line_lwd)),
  t = header_bottom, b = header_bottom, l = left_col, r = right_col,
  clip = "off", name = "header-bottom")

# Bottom line below last study
traffic_gt <- gtable::gtable_add_grob(traffic_gt,
  grid::segmentsGrob(x0 = 0, x1 = 1,
    y0 = unit(0, "npc") - gap_bot, y1 = unit(0, "npc") - gap_bot,
    gp = grid::gpar(col = line_col, lwd = line_lwd)),
  t = bottom_row, b = bottom_row, l = left_col, r = right_col,
  clip = "off", name = "bottom-line")

# Vertical separator between study labels and D1 (extended down to meet bottom line)
traffic_gt <- gtable::gtable_add_grob(traffic_gt,
  grid::segmentsGrob(x0 = 0, x1 = 0, y0 = unit(0, "npc") - gap_bot, y1 = 1,
    gp = grid::gpar(col = line_col, lwd = line_lwd)),
  t = top_row, b = bottom_row, l = sep_col, r = sep_col,
  clip = "off", name = "vert-separator")

grid::grid.newpage()
grid::grid.draw(traffic_gt)

```

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 (bar plot) of risk of bias assessments
rob_summary_plot <- rob_summary(rob_d, tool = "ROB2", weighted = FALSE)

# Replace full domain names with D1-D5 and 0
levels(rob_summary_plot$data$domain) <- c(
  "Weights" = "Weights",
  "Overall risk of bias" = "OA",
  "Bias in selection of the reported result" = "D5",
  "Bias in measurement of the outcome" = "D4",
  "Bias due to missing outcome data" = "D3",

```

	D1	D2	D3	D4	D5	OA
Chen et al. (2022)	⊖	⊕	⊖	⊕	⊕	⊖
Dubuson et al. (2021)	⊕	⊕	⊖	⊕	⊕	⊖
Eberl et al. (2013)	⊖	⊕	⊗	⊗	⊖	⊗
Garfield et al. (2022)*	⊕	⊕	⊖	⊕	⊕	⊖
Garfield et al. (2026)	⊕	⊕	⊖	⊕	⊕	⊖
Grieder et al. (2022)	⊕	⊖	⊖	⊕	⊕	⊖
Kumar et al. (2019)	⊗	⊖	⊖	⊖	⊕	⊗
Kvamme et al. (2019)	⊕	⊖	⊕	⊗	⊖	⊗
Laurens et al. (2023)	⊖	⊕	⊗	⊕	⊕	⊗
Manning et al. (2016)	⊕	⊕	⊕	⊕	⊕	⊕
Manning et al. (2021)*	⊕	⊕	⊕	⊕	⊕	⊕
Manning et al. (2022)	⊕	⊕	⊕	⊕	⊕	⊕
Mistarz et al. (2025)	⊕	⊕	⊗	⊕	⊕	⊗
Peerenboom (2022)	⊕	⊕	⊖	⊖	⊖	⊗
Peerenboom et al. (2026)*	⊕	⊕	⊖	⊖	⊖	⊗
Rinck et al. (2018)	⊕	⊕	⊕	⊕	⊕	⊕
Salemink et al. (2022)	⊖	⊕	⊕	⊖	⊕	⊖
Schenkel et al. (2023)	⊕	⊕	⊕	⊕	⊖	⊖
Schenkel et al. (2024)	⊕	⊕	⊖	⊖	⊖	⊗
Schoenmakers et al. (2010)	⊕	⊕	⊖	⊕	⊖	⊖
Spruyt et al. (2025)	⊕	⊖	⊕	⊕	⊕	⊖
Stein et al. (2023)	⊕	⊕	⊕	⊕	⊕	⊕
Uyl et al. (2018)	⊕	⊕	⊗	⊕	⊕	⊗
Wiers et al. (2011)	⊖	⊕	⊕	⊖	⊖	⊖

Judgement	
⊗	High risk
⊖	Some concerns
⊕	Low risk

Figure 1: Traffic plot of individual risk of bias ratings

```

"Bias due to deviations from intended interventions" = "D2",
"Bias arising from the randomization process"       = "D1",
"Study"                                              = "Study"
)

# Custom styling
rob_sum_styled <- rob_summary_plot +
  theme(
    axis.text.y    = element_text(face = "bold", size = 13, margin = margin(r = 10)),
    axis.text.x    = element_text(size = 13),
    axis.line.y    = element_line(colour = "black", linewidth = 0.4),
    legend.text    = element_text(size = 12),
    legend.title   = element_text(size = 12),
    legend.position = "bottom",
    legend.direction = "horizontal",
    legend.justification = "center",
    legend.background = element_rect(fill = "white", colour = "black",
                                     linetype = "solid", linewidth = 0.4),
    legend.margin   = margin(6, 6, 6, 6),
    legend.box.margin = margin(5, 0, 5, 0)
  ) +
  # reverse = TRUE: Low risk first, High risk last; weighted bar plot will be created later
  guides(fill = guide_legend(title = NULL, nrow = 1, reverse = TRUE))

# Stretch the legend box to span the full figure width via gtable
rob_sum_gt <- ggplotGrob(rob_sum_styled)

leg_nms <- intersect(
  rob_sum_gt$layout$name,
  c("guide-box", "guide-box-bottom", "guide-box-top",
    "guide-box-left", "guide-box-right", "guide-box-inside")
)

# Span full figure width (column 1 to last column)
l_col <- 1L
r_col <- ncol(rob_sum_gt)

for (nm in leg_nms) {
  idx <- which(rob_sum_gt$layout$name == nm)
  t_row <- rob_sum_gt$layout$t[idx]
  b_row <- rob_sum_gt$layout$b[idx]

  rob_sum_gt$layout$l[idx] <- l_col
  rob_sum_gt$layout$r[idx] <- r_col

  # Full-width background rectangle drawn behind the legend content
  rob_sum_gt <- gtable::gtable_add_grob(
    rob_sum_gt,
    grid::rectGrob(gp = grid::gpar(fill = "white", col = "black",
                                     lwd = 0.4 * ggplot2::.pt)),
    t = t_row, b = b_row, l = l_col, r = r_col,
    name = "legend-bg-full", z = min(rob_sum_gt$layout$z) - 1
  )
}

```

```
}
# Display the plot
grid.newpage()
grid.draw(rob_sum_gt)
```

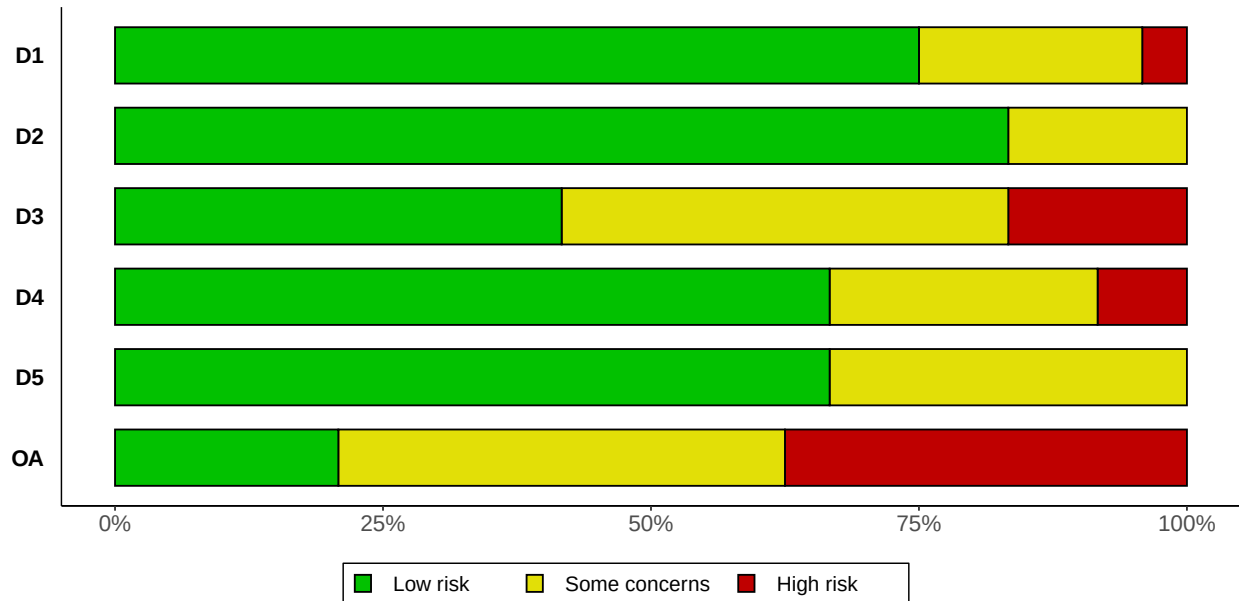


Figure 2: Summary of risk of bias ratings

2. Main analyses on short-term outcomes

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

2.1 Short-term abstinence

As we are starting with our main outcome of abstinence, 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 "abs_d" and preserve original order of studies
abs_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):

$$\ln OR = SMD \frac{\pi}{\sqrt{3}}$$

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

$$SMD = \ln OR \frac{\sqrt{3}}{\pi}$$

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 main 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
abs_d_3m <- abs_d %>% filter(time_point == "0-3 months")
```

The calculation of effect sizes will be done by the inbuilt functions in the `meta` package (Balduzzi et al. 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
abs_d_cont_3m <- abs_d_3m %>% filter(!is.na(e_abs_trans_mean))

# Filter studies only with dichotomous outcomes and store it as a new dataset
abs_d_cat_3m <- abs_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()
abs_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 = abs_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 abstinent, 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
abs_d_sumcont_3m <- data.frame(
  row_id = abs_d_cont_3m$row_id,
  study = abs_metacont_3m$studlab,
  time_point = abs_d_cont_3m$time_point,
  system = abs_d_cont_3m$system,
  intervention = abs_d_cont_3m$intervention,
  total_n = abs_d_cont_3m$e_n + abs_d_cont_3m$c_n, # Calculating total number of participants
  c_n = abs_d_cont_3m$c_n,
  c_abs = NA,
  c_non_abs = NA,
  e_n = abs_d_cont_3m$e_n,
  e_abs = NA,
  e_non_abs = NA,
  # Adding calculated SMDs to the data frame
  SMD = abs_metacont_3m$TE,
  SE = abs_metacont_3m$seTE,
  LCI = abs_metacont_3m$lower,
  UCI = abs_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
abs_d_sumcont_3m <- abs_d_sumcont_3m %>%
  data.frame(
    lnOR = abs_metacont_3m$TE * chinn_smd_to_lnor,
    SElnOR = abs_metacont_3m$seTE * chinn_smd_to_lnor,
    lnLCI = abs_metacont_3m$lower * chinn_smd_to_lnor,
    lnUCI = abs_metacont_3m$upper * chinn_smd_to_lnor,
    # Saving the ORs and its corresponding SE and 95% CI
    OR = exp(abs_metacont_3m$TE * chinn_smd_to_lnor),
    OrSE = exp(abs_metacont_3m$TE * chinn_smd_to_lnor) * (abs_metacont_3m$seTE * chinn_smd_to_lnor),
    OrLCI = exp(abs_metacont_3m$lower * chinn_smd_to_lnor),
    OrUCI = exp(abs_metacont_3m$upper * chinn_smd_to_lnor),
    # Calculating the z- and p-values
    `z-value` = (abs_metacont_3m$TE * chinn_smd_to_lnor) / (abs_metacont_3m$seTE * chinn_smd_to_lnor),
    `p-value` = 2 * (1 - pnorm(abs((abs_metacont_3m$TE * chinn_smd_to_lnor) / (abs_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()
abs_metabin_3m <- metabin(
  event.e = e_abs,
  n.e = e_n,
  event.c = c_abs,
  n.c = c_n,
  studlab = study,
  data = abs_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 abstinent, 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
abs_d_sumbin_3m <- data.frame(
  row_id = abs_d_cat_3m$row_id,
  study = abs_metabin_3m$studlab,
  time_point = abs_d_cat_3m$time_point,
  system = abs_d_cat_3m$system,
  intervention = abs_d_cat_3m$intervention,
  total_n = abs_d_cat_3m$e_n + abs_d_cat_3m$c_n,
  c_n = abs_d_cat_3m$c_n,
  c_abs = abs_d_cat_3m$c_abs,
  c_non_abs = abs_d_cat_3m$c_n - abs_d_cat_3m$c_abs,
  e_n = abs_d_cat_3m$e_n,
  e_abs = abs_d_cat_3m$e_abs,
  e_non_abs = abs_d_cat_3m$e_n - abs_d_cat_3m$e_abs,
# Extracting the lnORs from the metabin results
  lnOR = abs_metabin_3m$TE,
  SElnOR = abs_metabin_3m$seTE,
  lnLCI = abs_metabin_3m$lower,
  lnUCI = abs_metabin_3m$upper,
# Calculating the ORs from the metabin results
  OR = exp(abs_metabin_3m$TE),
  OrSE = exp(abs_metabin_3m$TE) * abs_metabin_3m$seTE,
  OrLCI = exp(abs_metabin_3m$lower),
  OrUCI = exp(abs_metabin_3m$upper),
# Calculating the SMD, SE and 95% CI using the Chinn-function
  SMD = abs_metabin_3m$TE * chinn_lnor_to_smd,
  SE = abs_metabin_3m$seTE * chinn_lnor_to_smd,
  LCI = abs_metabin_3m$lower * chinn_lnor_to_smd,
  UCI = abs_metabin_3m$upper * chinn_lnor_to_smd,
# Adding the z- and p-values
  `z-value` = abs_metabin_3m$TE / abs_metabin_3m$seTE,
```

```

`p-value` = 2 * (1 - pnorm(abs(abs_metabin_3m$TE / abs_metabin_3m$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 abs and round all the values 5 decimals
abs_d_full_3m <- bind_rows(abs_d_sumcont_3m, abs_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 of short-term effects on abstinence

The short-term effects (0-3 months) on abstinence has now been calculated, and the studies with missing categorical outcomes have been converted, so our analysis now includes 12 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_2026" = "Garfield et al. (2026)",
  "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)",
"17a_stein_2023" = "Stein et al. (2023)",
"17b_stein_2023" = "Stein et al. (2023)",
"18a_wiers_2011" = "Wiers et al. (2011)",
"18b_wiers_2011" = "Wiers et al. (2011)",
"19a_spruyt_2025" = "Spruyt et al. (2025)",
"19b_spruyt_2025" = "Spruyt et al. (2025)",
"19c_spruyt_2025" = "Spruyt et al. (2025)",
"20a_mistarz_2025" = "Mistarz et al. (2025)",
"20b_mistarz_2025" = "Mistarz et al. (2025)",
"21_salemink_2022" = "Salemink et al. (2022)"
)

```

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

```

# Assign the new study labels to the combined data frame
abs_d_full_3m$study <- ifelse(abs_d_full_3m$study %in% names(study_labels),
                             study_labels[abs_d_full_3m$study],
                             abs_d_full_3m$study)

```

We want to have the author labels 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
abs_d_full_3m <- abs_d_full_3m %>%
  arrange(study)

```

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
abs_d_full_3m <- abs_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
abs_d_full_3m <- abs_d_full_3m %>%
  dplyr::rename(`Cognitive system` = system) %>%
  dplyr::mutate( # Changing the order of the two subgroups
    `Cognitive system` = factor(`Cognitive system`,
                               levels = c("Implicit", "Explicit"))
  )

```

With the new subgroup names, we can pool the effects across studies and stratify by the cognitive systems targeted by the intervention 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 cognitive systems and using the new labels
abs_metagen_3m <- metagen(TE = lnOR,
                          seTE = SElnOR,
                          studlab = study,
                          data = abs_d_full_3m,
                          subgroup = `Cognitive system`,

```

```

sm = "OR",
method.tau = "REML",
common = FALSE,
random = TRUE,
method.random.ci = "classic")

# Print the results of the meta-analysis
print(abs_metagen_3m)

## Number of studies: k = 13
##
##              OR          95% CI    z p-value
## Random effects model 1.2430 [1.0789; 1.4321] 3.01 0.0026
##
## Quantifying heterogeneity (with 95% CIs):
## tau^2 < 0.0001 [0.0000; 0.8164]; tau = 0.0016 [0.0000; 0.9035]
## I^2 = 32.6% [0.0%; 65.2%]; H = 1.22 [1.00; 1.69]
##
## Test of heterogeneity:
##      Q d.f. p-value
## 17.80  12  0.1218
##
## Results for subgroups (random effects model):
##              k      OR          95% CI  tau^2    tau    Q
## Cognitive system = Implicit 11 1.2025 [1.0412; 1.3889]    0    0 9.46
## Cognitive system = Explicit  2 3.5026 [1.0190; 12.0390] 0.4747 0.6890 2.44
##              I^2
## Cognitive system = Implicit 0.0%
## Cognitive system = Explicit 59.0%
##
## Test for subgroup differences (random effects model):
##              Q d.f. p-value
## Between groups 2.84    1 0.0919
##
## 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

```

Then we create a helper function for the calculation of the number needed to treat, which can be used for the forest plots.

```

# NNT from OR and control abstinence rate
nnt <- function(OR, p_c) {
  p_e <- (OR * p_c) / (1 - p_c + OR * p_c) # experimental abstinence rate
  risk_dif <- p_e - p_c # risk difference
  ifelse(is.na(risk_dif) | risk_dif == 0, NA, 1 / risk_dif)
}

```

We can now calculate the baseline control risk (assumed comparator risk [ACR]) for the short-term effects by only including studies with categorical outcomes.

```

# Control abstinence rate risk (assumed control risk) across all studies on short-term effects
p_c_abs_3m <- with(abs_d_sumbin_3m, sum(c_abs) / sum(c_n))

```

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
abs_zp_overall_3m <- data.frame(
  lnOR = abs_metagen_3m$TE.random,
  lnOR_se = abs_metagen_3m$seTE.random,
  NNT = round(nnt(exp(abs_metagen_3m$TE.random), p_c_abs_3m)),
  z_value = abs_metagen_3m$statistic.random,
  p_value = abs_metagen_3m$pval.random
)

# Extract the test statistics for the subgroup analysis and create a data frame
abs_zp_subgroup_3m <- data.frame(
  subgroup = abs_metagen_3m$bylevs,
  lnOR = abs_metagen_3m$TE.random.w,
  lnOR_se = abs_metagen_3m$seTE.random.w,
  NNT = round(nnt(exp(abs_metagen_3m$TE.random.w), p_c_abs_3m)),
  z_value = abs_metagen_3m$statistic.random.w,
  p_value = abs_metagen_3m$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 cognitive system
abs_zp_subsplit_3m <- split(abs_zp_subgroup_3m, abs_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., abstinent), 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 I^2 .

```
# Now plot the forest plot again with new subgroup names
forest(abs_metagen_3m,
  prediction = TRUE,
  prediction.subgroup = FALSE,
  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,
    xlim = c(1/30, 30),
    at = c(0.03, 0.08, 0.2, 0.5, 1, 2, 5, 12, 30),
    leftcols = c("studlab", "e_abs", "e_n", "c_abs", "c_n"),
    leftlabs = c("Study", "Intv. (events)", "Total (n)", "Ctrl. (events)", "Total (n)"),
    rightcols = c("w.random", "effect", "ci"),
    rightlabs = c("Weight (%)", "OR", "95% CI")
  )

# Add a footer for the test statistics of the effects for implicit system
footer_imp_abs_3m <- paste0(
  "Subgroup effect",
  paste0(
    ": z = ", round(abs_zp_subsplit_3m$Implicit$z_value, 2),
    ", p = ", formatC(abs_zp_subsplit_3m$Implicit$p_value, digits = 2),
    ", NNT = ", abs_zp_subsplit_3m$Implicit$NNT
  )
)

# Add a footer for the test statistics of the effects for explicit system
footer_exp_abs_3m <- paste0(
  "Subgroup effect",
  paste0(
    ": z = ", round(abs_zp_subsplit_3m$Explicit$z_value, 2),
    ", p = ", formatC(abs_zp_subsplit_3m$Explicit$p_value, digits = 2),
    ", NNT = ", abs_zp_subsplit_3m$Explicit$NNT
  )
)

footer_over_abs_3m <- paste0(
  "Overall effect: ",
  "z = ", round(abs_zp_overall_3m$z_value, 2),
  ", p = ", formatC(abs_zp_overall_3m$p_value, digits = 2),
  ", NNT = ", abs_zp_overall_3m$NNT
)

# Adjust the footer for the subgroup effects after implicit subgroup
grid.text(footer_imp_abs_3m, x = 0.079, y = 0.38,
  just = "left", gp = gpar(fontsize = 10))

# Adjust the footer for the subgroup effects after explicit subgroup
grid.text(footer_exp_abs_3m, x = 0.079, y = 0.2,
  just = "left", gp = gpar(fontsize = 10))

# Adjust the footer for the overall effect to the bottom left
grid.text(footer_over_abs_3m, x = 0.079, y = 0.08,
  just = "left", gp = gpar(fontsize = 10))

```

If we want to know the prediction intervals without outputting it to the plot, we can print them separately.

```

# Displaying the log-transformed prediction interval (PI) from the metagen output and rounding the output
round(
  exp(cbind(

```

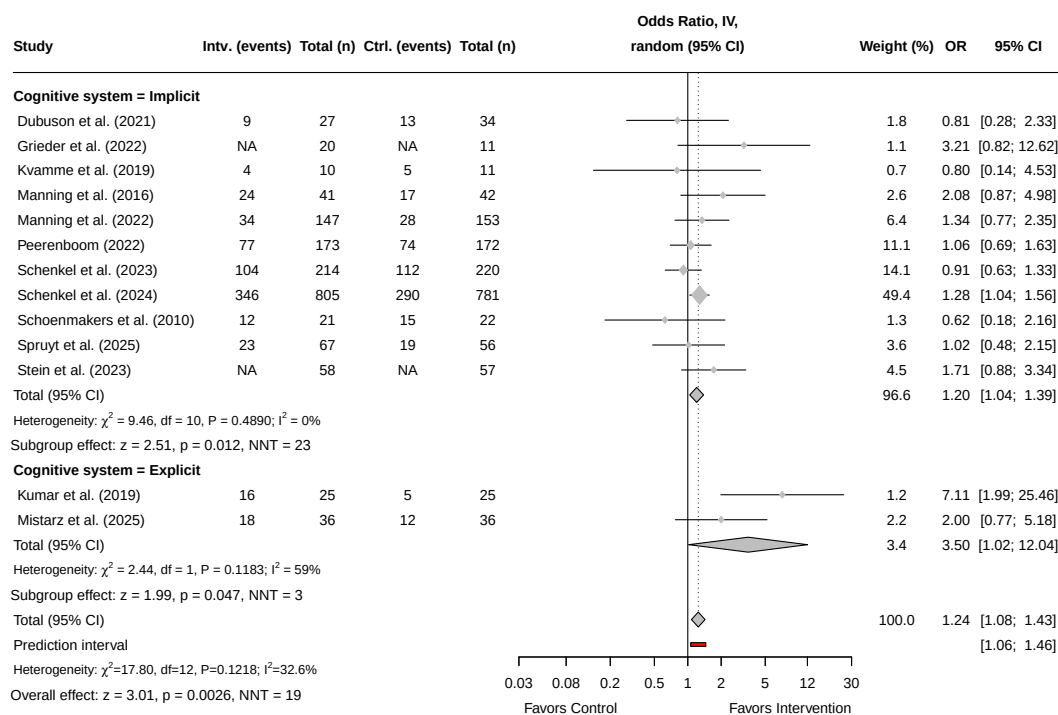


Figure 3: Forest plot for the effects on abstinence

```

    LL = abs_metagen_3m$lower.predict.w,
    UL = abs_metagen_3m$upper.predict.w
  )
), 5
)

```

```

##           LL           UL
## Implicit 1.02083      1.41655
## Explicit 0.00002 496531.65336

```

2.1.5 Sensitivity analysis for short-term effects on abstinence

In the main analysis, we included two studies that did not report a categorical outcome for 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
abs_d_sens_filter_3m <- abs_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 than by the pre-defined row numbers.

```

# Arrange studies alphabetically
abs_d_sens_filter_3m <- abs_d_sens_filter_3m %>%
  arrange(study)

```

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 cognitive systems
abs_sens_metagen_3m <- metagen(TE = lnOR,
  seTE = SElnOR,
  studlab = study,
  data = abs_d_sens_filter_3m,
  subgroup = `Cognitive system`,
  sm = "OR",
  method.tau = "REML",
  common = FALSE,
  random = TRUE,
  method.random.ci = "classic")

# Print the results of the meta-analysis
print(abs_sens_metagen_3m)

```

```

## Number of studies: k = 11
##
##           OR           95% CI      z p-value
## Random effects model 1.2089 [1.0381; 1.4078] 2.44 0.0147
##
## Quantifying heterogeneity (with 95% CIs):
## tau^2 = 0.0017 [0.0000; 1.0095]; tau = 0.0410 [0.0000; 1.0047]
## I^2 = 33.1% [0.0%; 67.1%]; H = 1.22 [1.00; 1.74]
##
## Test of heterogeneity:
##      Q d.f. p-value

```

```
## 14.95    10  0.1339
##
## Results for subgroups (random effects model):
##           k      OR          95% CI   tau^2    tau    Q
## Cognitive system = Implicit  9 1.1568 [0.9839; 1.3601] 0.0034 0.0585 6.26
## Cognitive system = Explicit  2 3.5026 [1.0190; 12.0390] 0.4747 0.6890 2.44
##           I^2
## Cognitive system = Implicit  0.0%
## Cognitive system = Explicit 59.0%
##
## Test for subgroup differences (random effects model):
##           Q d.f. p-value
## Between groups 3.04    1  0.0812
##
## 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
abs_zp_sens_over_3m <- data.frame(
  lnOR = abs_sens_metagen_3m$TE.random,
  lnOR_se = abs_sens_metagen_3m$seTE.random,
  NNT = round(nnt(exp(abs_sens_metagen_3m$TE.random), p_c_abs_3m)),
  z_value = abs_sens_metagen_3m$statistic.random,
  p_value = abs_sens_metagen_3m$pval.random
)

# Extract the test statistics for the subgroup analysis and create a data frame
abs_zp_sens_sub_3m <- data.frame(
  subgroup = abs_sens_metagen_3m$bylevs,
  lnOR = abs_sens_metagen_3m$TE.random.w,
  lnOR_se = abs_sens_metagen_3m$seTE.random.w,
  NNT = round(nnt(exp(abs_sens_metagen_3m$TE.random.w), p_c_abs_3m)),
  z_value = abs_sens_metagen_3m$statistic.random.w,
  p_value = abs_sens_metagen_3m$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 cognitive system
abs_zp_sens_subsplit_3m <- split(abs_zp_sens_sub_3m, abs_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(abs_sens_metagen_3m,
  prediction = TRUE,
  prediction.subgroup = FALSE,
  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,
xlim = c(1/30, 30),
at = c(0.03, 0.08, 0.2, 0.5, 1, 2, 5, 12, 30),
leftcols = c("studlab", "e_abs", "e_n", "c_abs", "c_n"),
leftlabs = c("Study", "Intv. (events)", "Total (n)", "Ctrl. (events)", "Total (n)"),
# Right side: default BMJ columns
rightcols = c("w.random", "effect", "ci"),
rightlabs = c("Weight (%)", "OR", "95% CI")
)

# Add a footer for the test statistics of the effects for implicit system
footer_imp_abs_sens_3m <- paste0(
  "Subgroup effect",
  paste0(
    ": z = ", round(abs_zp_sens_subsplit_3m$Implicit$z_value, 2),
    ", p = ", formatC(abs_zp_sens_subsplit_3m$Implicit$p_value, digits = 2),
    ", NNT = ", abs_zp_sens_subsplit_3m$Implicit$NNT
  )
)

# Add a footer for the test statistics of the effects for explicit system
footer_exp_abs_sens_3m <- paste0(
  "Subgroup effect",
  paste0(
    ": z = ", round(abs_zp_sens_subsplit_3m$Explicit$z_value, 2),
    ", p = ", formatC(abs_zp_sens_subsplit_3m$Explicit$p_value, digits = 2),
    ", NNT = ", abs_zp_sens_subsplit_3m$Explicit$NNT
  )
)

footer_over_sens_abs_3m <- paste0(
  "Overall effect: ",
  "z = ", round(abs_zp_sens_over_3m$z_value, 2),
  ", p = ", formatC(abs_zp_sens_over_3m$p_value, digits = 2),
  ", NNT = ", abs_zp_sens_over_3m$NNT
)

# Adjust the footer for the subgroup effects after implicit subgroup
grid.text(footer_imp_abs_sens_3m, x = 0.079, y = 0.4,

```

```

just = "left", gp = gpar(fontsize = 10))

# Adjust the footer for the subgroup effects after explicit subgroup
grid.text(footer_exp_abs_sens_3m, x = 0.079, y = 0.19,
just = "left", gp = gpar(fontsize = 10))

# Adjust the footer for the overall effect to the bottom left
grid.text(footer_over_sens_abs_3m, x = 0.079, y = 0.05,
just = "left", gp = gpar(fontsize = 10))

```

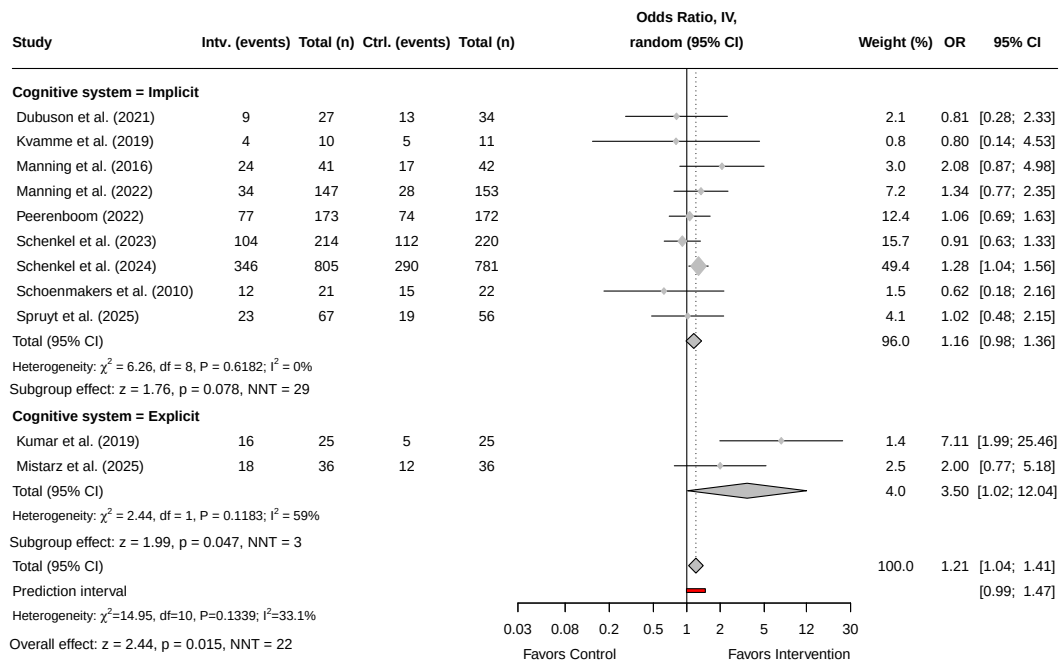


Figure 4: Forest plot for the sensitivity analysis on abstinence

```

# Displaying the log-transformed prediction interval (PI) from the metagen output and rounding the output
round(
  exp(cbind(
    LL = abs_sens_metagen_3m$lower.predict.w,
    UL = abs_sens_metagen_3m$upper.predict.w
  )), 5
)

```

```

##          LL          UL
## Implicit 0.91599      1.46103
## Explicit 0.00002 496531.65336

```

2.1.6 Publication bias for short-term effects on abstinence

The final step for the main analyses on 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 cognitive systems. 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
abs_pupbias_3m <- metagen(TE = lnOR,
                          seTE = SElnOR,
                          studlab = study,
                          sm = "OR",
                          method.tau = "REML",
                          data = abs_d_full_3m)

# Print the results of the meta-analysis without subgroups
print(abs_pupbias_3m)

## Number of studies: k = 13
##
##              OR          95% CI      z p-value
## Common effect model  1.2430 [1.0789; 1.4321] 3.01  0.0026
## Random effects model 1.2430 [1.0789; 1.4321] 3.01  0.0026
##
## Quantifying heterogeneity (with 95% CIs):
## tau^2 < 0.0001 [0.0000; 0.8164]; tau = 0.0016 [0.0000; 0.9035]
## I^2 = 32.6% [0.0%; 65.2%]; H = 1.22 [1.00; 1.69]
##
## Test of heterogeneity:
##      Q d.f. p-value
## 17.80  12  0.1218
##
## 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 previously calculated treatment effects
abs_egger_3m <- metabias(abs_pupbias_3m, method.bias = "Egger")

# Print the test results of the Egger's test
print(abs_egger_3m)
```

```
## Linear regression test of funnel plot asymmetry
##
## Test result: t = 0.92, df = 11, p-value = 0.3768
## Bias estimate: 0.5337 (SE = 0.5795)
##
## Details:
## - multiplicative residual heterogeneity variance (tau^2 = 1.5026)
## - 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(abs_pupbias_3m,
       xlab = "Odds Ratio",
       studlab = TRUE,
       backtransf = TRUE,
       pch = 17,
       cex = 1)

# Add Egger's test statistic to the plot
abs_egger_text_3m <- paste0("Egger's test: p = ",
                             round(abs_egger_3m$p.value, 3))

# Define the actual placement of the statistics on the funnel plot
mtext(abs_egger_text_3m, side=3, line=0.5, adj=0, cex=0.9, col="black")
```

2.2 Short-term alcohol reduction

The first part of the main analysis was on our primary outcome abstinence, now we can turn to analyzing 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 "abs_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 main 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")
```

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",
```

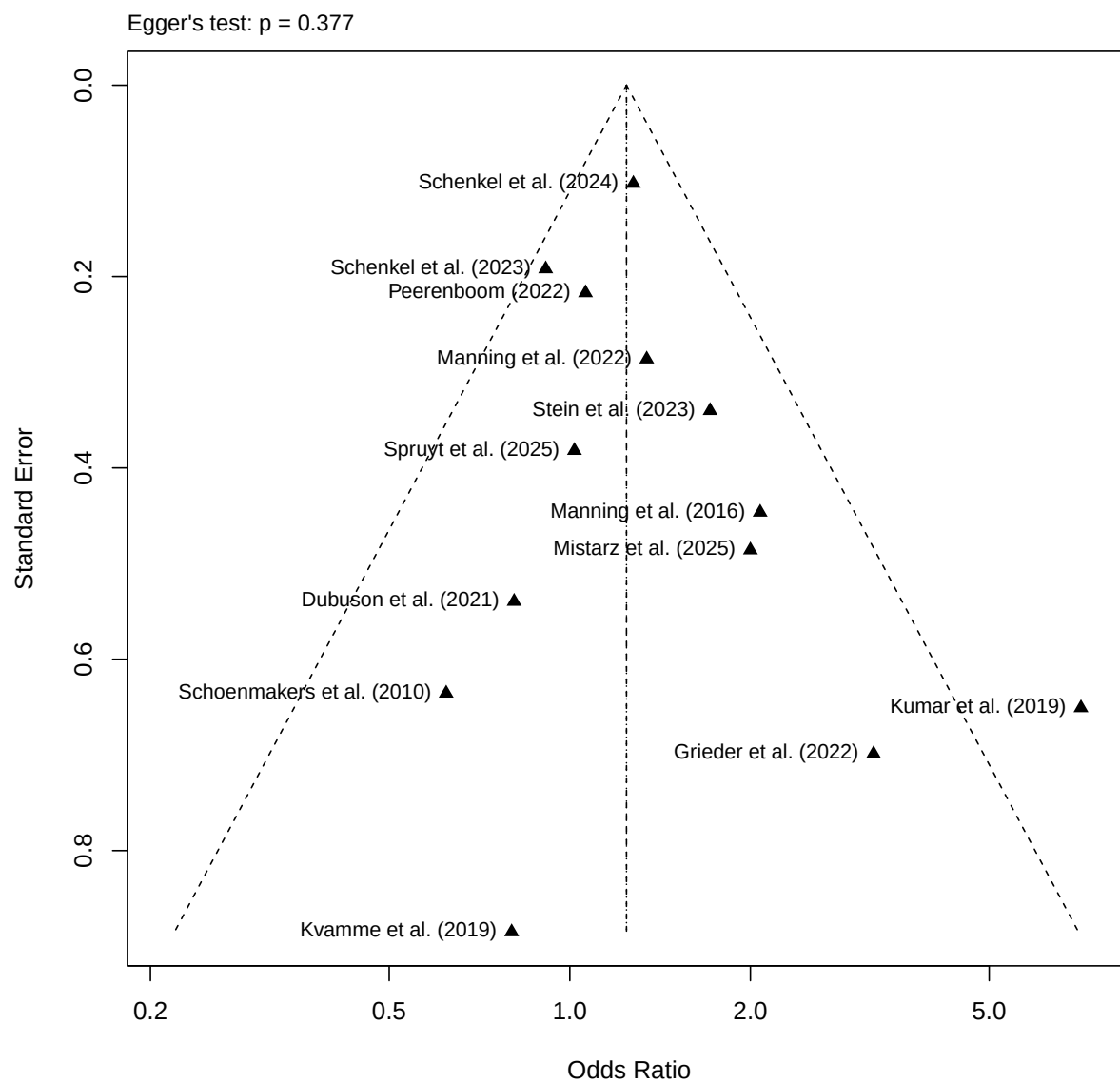


Figure 5: Funnel plot for studies on short-term effects on abstinence

```

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$time_point,
  system = red_d_3m$system,
  intervention = red_d_3m$intervention,
  total_n = red_d_3m$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$e_reduc_sd,
  # Adding calculated SMDs to the data frame
  SMD = red_metacont_3m$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 of short-term effects 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)

```

We want to have the author labels 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
red_d_full_3m <- red_d_full_3m %>%
  arrange(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 interventions on alcohol reduction, we do not apply stratification by subgroups.

```

# Run the meta-analysis stratified by cognitive systems 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 abs, however, no subgroup labels will be present, as only effects of cognitive bias modification programs 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))

```

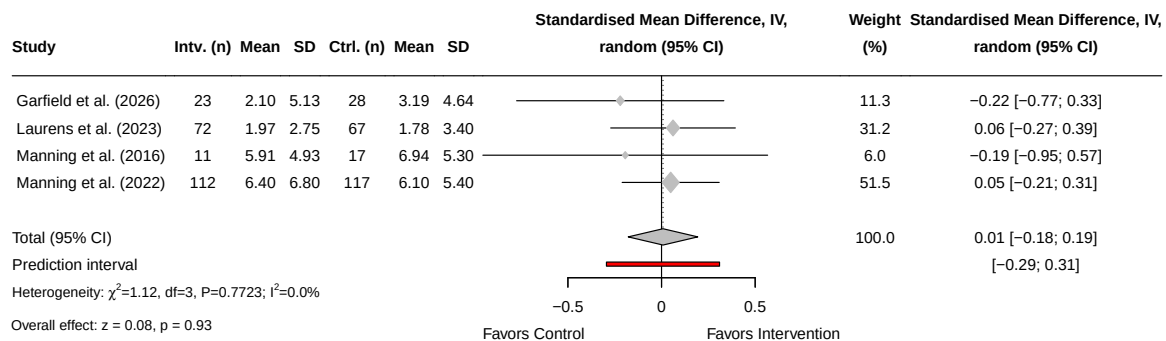


Figure 6: Forest plot of effects on alcohol reduction

2.2.3 Publication bias for short-term effects on alcohol reduction

The final step for the main 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")
```

2.3 Short-term alcohol craving

The final part of the main analyses 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 craving 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 main 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.

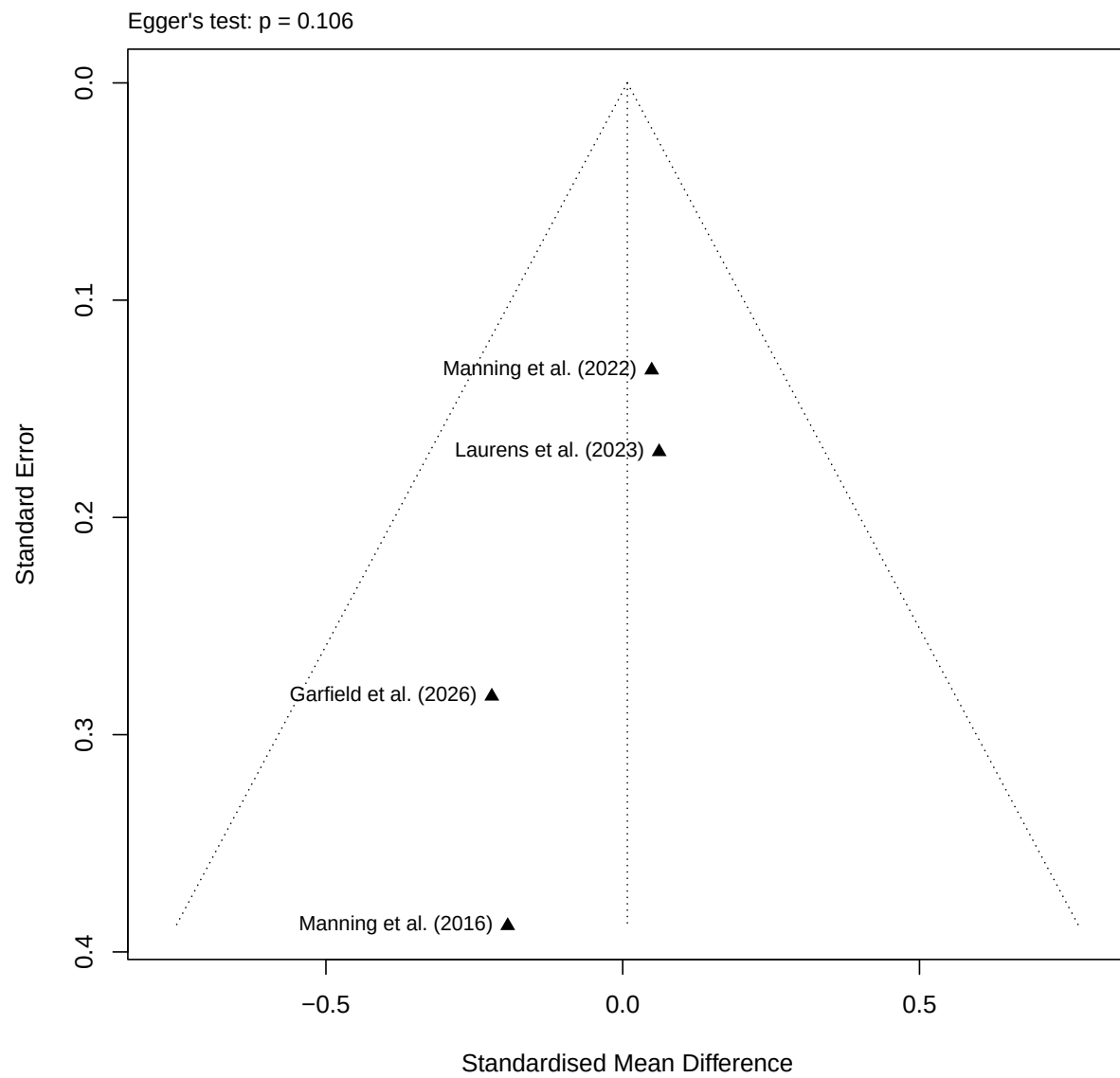


Figure 7: Funnel plot on studies for short-term effects on alcohol reduction

```
# Filter 0-3 month data and arrange specific studies first
crav_d_3m <- crav_d %>%
  filter(time_point == "0-3 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_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$time_point,
  system = crav_d_3m$system,
  intervention = crav_d_3m$intervention,
  total_n = crav_d_3m$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$e_crav_sd,
  # Adding calculated SMDs to the data frame
  SMD = crav_metacont_3m$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 of short-term effects 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)
```

We want to have the author labels 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
crav_d_full_3m <- crav_d_full_3m %>%
  arrange(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 analysis. However, as no study examined the effects of explicit interventions on alcohol craving, we do not apply stratification by subgroups.

```
# Run the meta-analysis stratified by cognitive systems 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 interventions on craving 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))
```

2.3.3 Publication bias for short-term effects on craving

The final step for the main 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

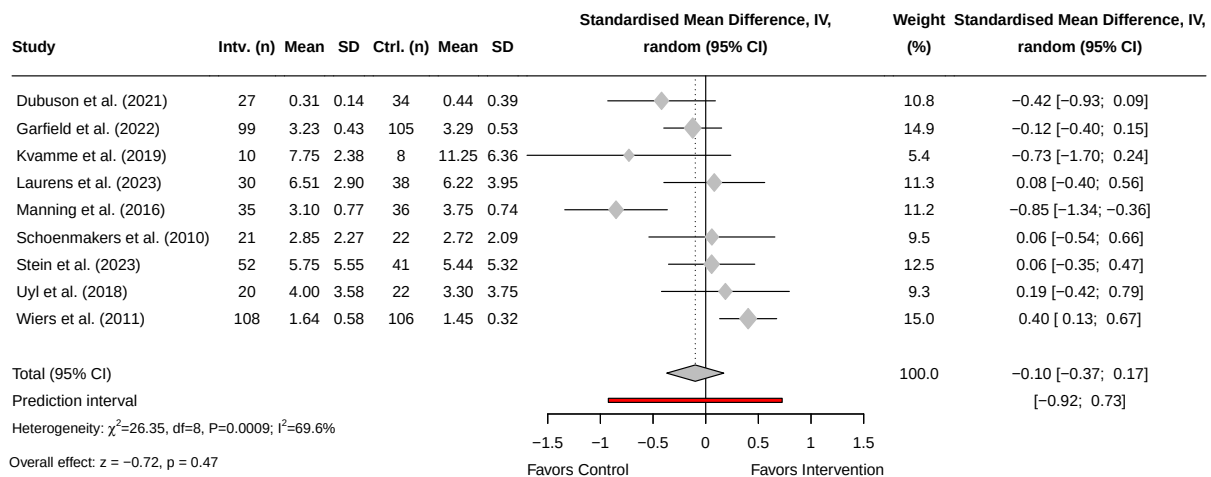


Figure 8: Forest plot of the effects on craving

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")

```

3. Main analyses on intermediate-term outcomes

The second part of the main analyses will be conducted on the intermediate time points. Intermediate effects will consist of studies examining intervention effects six 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

3.1 Intermediate-term abstinence

The first part be conducted on our primary outcome of abstinence, and we calculate the effect sizes for the intermediate-term time point.

3.1.1 Calculation of effect sizes for intermediate categorical variables

We want to follow the procedure of converting the continuous variables to categorical outcomes and in opposite direction as mentioned previously 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
abs_d_6m <- abs_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()
abs_metabin_6m <- metabin(
  event.e = e_abs,
  n.e = e_n,
  event.c = c_abs,
  n.c = c_n,
  studlab = study,
  data = abs_d_6m,
  sm = "OR",
  method.tau = "REML",
  method.random.ci = "classic",
  random = TRUE,
  incr = 0.5,

```

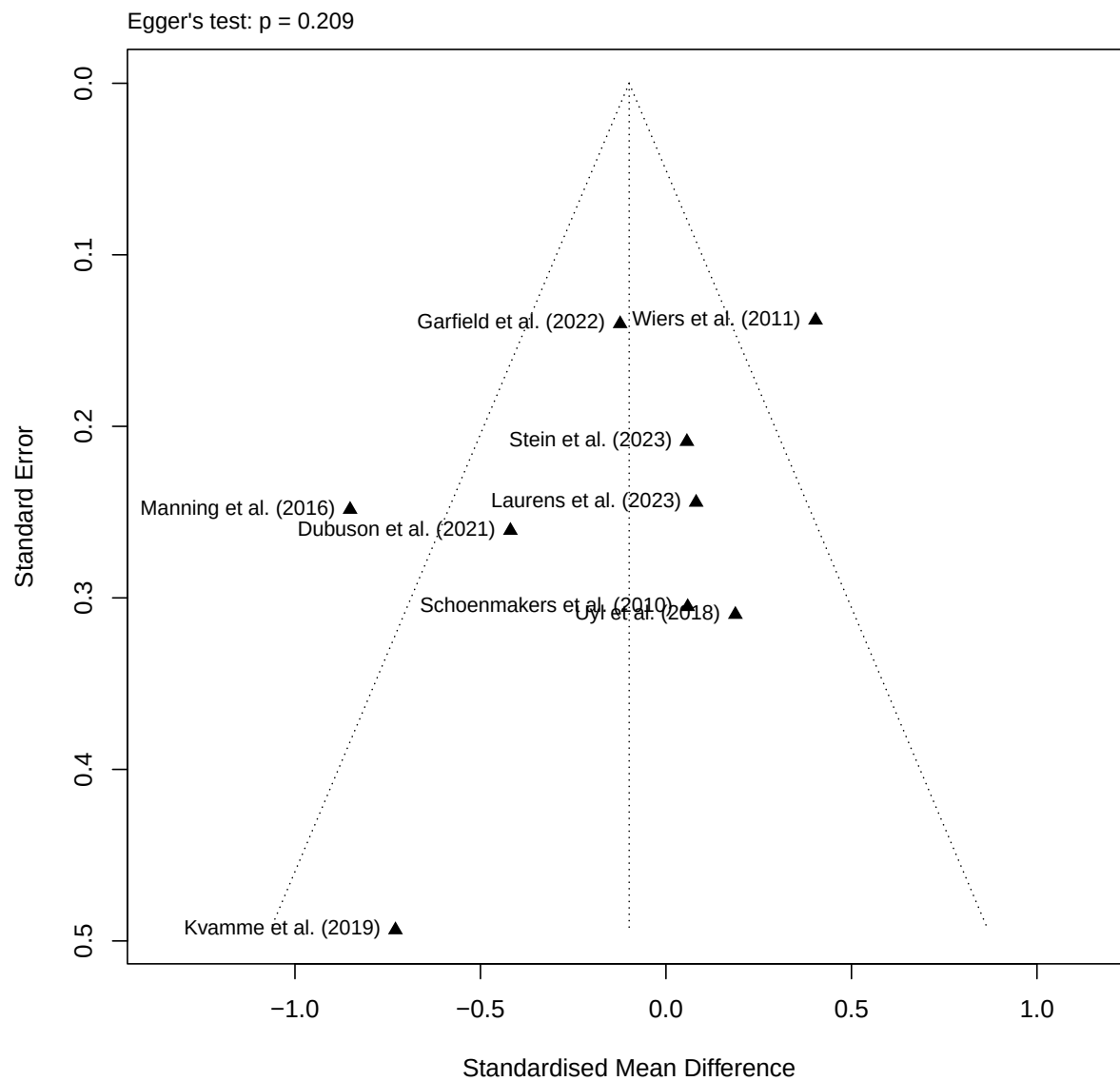


Figure 9: Funnel plot of the studies on short-term effects on craving

```

  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 abstinent, 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
abs_d_full_6m <- data.frame(
  row_id = abs_d_6m$row_id,
  study = abs_metabin_6m$studlab,
  time_point = abs_d_6m$time_point,
  system = abs_d_6m$system,
  intervention = abs_d_6m$intervention,
  total_n = abs_d_6m$e_n + abs_d_6m$c_n,
  c_n = abs_d_6m$c_n,
  c_abs = abs_d_6m$c_abs,
  c_non_abs = abs_d_6m$c_n - abs_d_6m$c_abs,
  e_n = abs_d_6m$e_n,
  e_abs = abs_d_6m$e_abs,
  e_non_abs = abs_d_6m$e_n - abs_d_6m$e_abs,
  # Extracting the lnORs from the metabin results
  lnOR = abs_metabin_6m$TE,
  SElnOR = abs_metabin_6m$seTE,
  lnLCI = abs_metabin_6m$lower,
  lnUCI = abs_metabin_6m$upper,
  # Calculating the ORs from the metabin results
  OR = exp(abs_metabin_6m$TE),
  OrSE = exp(abs_metabin_6m$TE) * abs_metabin_6m$seTE,
  OrLCI = exp(abs_metabin_6m$lower),
  OrUCI = exp(abs_metabin_6m$upper),
  # Calculating the SMD, SE and 95% CI using the Chinn-function
  SMD = abs_metabin_6m$TE * chinn_lnor_to_smd,
  SE = abs_metabin_6m$seTE * chinn_lnor_to_smd,
  LCI = abs_metabin_6m$lower * chinn_lnor_to_smd,
  UCI = abs_metabin_6m$upper * chinn_lnor_to_smd,
  # Adding the z- and p-values
  `z-value` = abs_metabin_6m$TE / abs_metabin_6m$seTE,
  `p-value` = 2 * (1 - pnorm(abs(abs_metabin_6m$TE / abs_metabin_6m$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
)

```

We want to have the APA author labels rather than study IDs.

```

# Assign the new study labels to the combined data frame
abs_d_full_6m$study <- ifelse(abs_d_full_6m$study %in% names(study_labels),
                             study_labels[abs_d_full_6m$study],
                             abs_d_full_6m$study)

```

We want to have the author labels 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
abs_d_full_6m <- abs_d_full_6m %>%
  arrange(study)
```

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
abs_d_full_6m <- abs_d_full_6m %>%
  dplyr::rename(`Cognitive system` = system) %>%
  dplyr::mutate( # Changing the order of the two subgroups
    `Cognitive system` = factor(`Cognitive system`,
      levels = c("Implicit", "Explicit"))
  )
```

With the new subgroup names, we can pool the effects across studies and stratify by the cognitive systems targeted by the intervention 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 cognitive systems and using the new labels
abs_metagen_6m <- metagen(TE = lnOR,
  seTE = SElnOR,
  studlab = study,
  data = abs_d_full_6m,
  subgroup = `Cognitive system`,
  sm = "OR",
  method.tau = "REML",
  common = FALSE,
  random = TRUE,
  method.random.ci = "classic")

# Print the results of the meta-analysis
print(abs_metagen_6m)
```

```
## Number of studies: k = 7
##
##              OR              95% CI      z p-value
## Random effects model 1.1406 [0.9478; 1.3726] 1.39  0.1639
##
## Quantifying heterogeneity (with 95% CIs):
## tau^2 < 0.0001 [0.0000; 1.7423]; tau = 0.0007 [0.0000; 1.3200]
## I^2 = 20.3% [0.0%; 63.8%]; H = 1.12 [1.00; 1.66]
##
## Test of heterogeneity:
##      Q d.f. p-value
## 7.53   6  0.2746
##
## Results for subgroups (random effects model):
##              k      OR              95% CI  tau^2    tau    Q
## Cognitive system = Implicit  5 1.0164 [0.7458; 1.3852] 0.0332 0.1821 4.66
## Cognitive system = Explicit  2 2.0899 [0.7808; 5.5937] 0.0892 0.2986 1.18
##
##              I^2
## Cognitive system = Implicit 14.2%
## Cognitive system = Explicit 14.9%
```

```
##
## Test for subgroup differences (random effects model):
##           Q d.f. p-value
## Between groups 1.87      1  0.1710
##
## 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
```

We can now calculate the baseline control risk for the intermediate-term effects by only including studies with categorical outcomes.

```
# Abstinence rate risk difference across all studies on short-term effects
p_c_abs_6m <- with(abs_d_full_6m, sum(c_abs) / sum(c_n))
```

3.1.2 Meta-analysis of intermediate-term effects on abstinence

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
abs_zp_overall_6m <- data.frame(
  lnOR = abs_metagen_6m$TE.random,
  lnOR_se = abs_metagen_6m$seTE.random,
  NNT = round(nnt(exp(abs_metagen_6m$TE.random), p_c_abs_6m)),
  z_value = abs_metagen_6m$statistic.random,
  p_value = abs_metagen_6m$pval.random
)

# Extract the test statistics for the subgroup analysis and create a data frame
abs_zp_subgroup_6m <- data.frame(
  subgroup = abs_metagen_6m$bylevs,
  lnOR = abs_metagen_6m$TE.random.w,
  lnOR_se = abs_metagen_6m$seTE.random.w,
  NNT = round(nnt(exp(abs_metagen_6m$TE.random.w), p_c_abs_6m)),
  z_value = abs_metagen_6m$statistic.random.w,
  p_value = abs_metagen_6m$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 cognitive system
abs_zp_subsplit_6m <- split(abs_zp_subgroup_6m, abs_zp_subgroup_6m$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., abstinent), 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 I^2 .

```
# Now plot the forest plot again with new subgroup names
forest(abs_metagen_6m,
       prediction = TRUE,
       prediction.subgroup = FALSE,
       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,
       xlim = c(1/30, 30),
       at = c(0.03, 0.08, 0.2, 0.5, 1, 2, 5, 12, 30),
       leftcols = c("studlab", "e_abs", "e_n", "c_abs", "c_n"),
       leftlabs = c("Study", "Intv. (events)", "Total (n)", "Ctrl. (events)", "Total (n)"),
       # Right side: default BMJ columns
       rightcols = c("w.random", "effect", "ci"),
       rightlabs = c("Weight (%)", "OR", "95% CI")
)

# Add a footer for the test statistics of the effects for implicit system
footer_imp_abs_6m <- paste0(
  "Subgroup effect ",
  paste0(
    ": z = ", round(abs_zp_subsplit_6m$Implicit$z_value, 2),
    ", p = ", formatC(abs_zp_subsplit_6m$Implicit$p_value, digits = 2),
    ", NNT = ", abs_zp_subsplit_6m$Implicit$NNT
  )
)

# Add a footer for the test statistics of the effects for explicit system
footer_exp_abs_6m <- paste0(
  "Subgroup effect",
  paste0(
    ": z = ", round(abs_zp_subsplit_6m$Explicit$z_value, 2),
    ", p = ", formatC(abs_zp_subsplit_6m$Explicit$p_value, digits = 2),
    ", NNT = ", abs_zp_subsplit_6m$Explicit$NNT
  )
)

footer_over_abs_6m <- paste0(
  "Overall effect: ",
  "z = ", round(abs_zp_overall_6m$z_value, 2),
```

```

", p = ", formatC(abs_zp_overall_6m$p_value, digits = 2),
", NNT = ", abs_zp_overall_6m$NNT
)

# Adjust the footer for the subgroup effects after implicit subgroup
grid.text(footer_imp_abs_6m, x = 0.094, y = 0.470,
          just = "left", gp = gpar(fontsize = 10))

# Adjust the footer for the subgroup effects after explicit subgroup
grid.text(footer_exp_abs_6m, x = 0.094, y = 0.230,
          just = "left", gp = gpar(fontsize = 10))

# Adjust the footer for the overall effect to the bottom left
grid.text(footer_over_abs_6m, x = 0.094, y = 0.060,
          just = "left", gp = gpar(fontsize = 10))

```

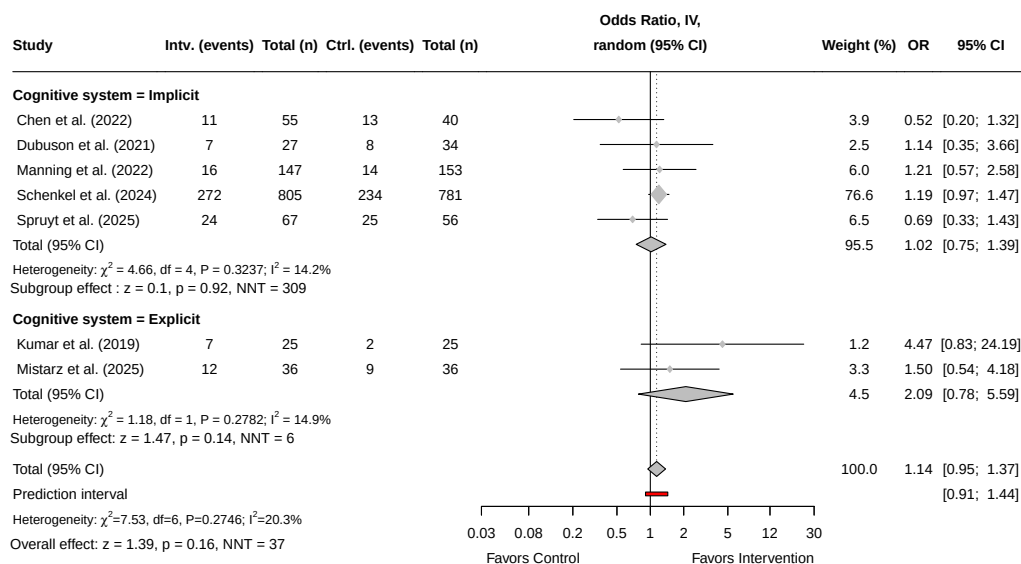


Figure 10: Forest plot for the intermediate-term effects on abstinence

If we want to know the prediction intervals without outputting it to the plot, we can print them separately.

```

# Displaying the log-transformed prediction interval (PI) from the metagen output and rounding the output
round(
  exp(cbind(
    LL = abs_metagen_6m$lower.predict.w,
    UL = abs_metagen_6m$upper.predict.w
  )), 5
)

##           LL           UL
## Implicit 0.52051    1.98485
## Explicit 0.00125   3506.08888

```

3.1.3 Publication bias for intermediate effects on abstinence

To assess for potential publication bias, 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.

```
# Assign the new study labels to the data frame
abs_d_full_6m$study <- ifelse(abs_d_full_6m$study %in% names(study_labels),
                              study_labels[abs_d_full_6m$study],
                              abs_d_full_6m$study)
```

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

```
# Run the meta-analysis stratified by cognitive systems and using the new labels
abs_metagen_6m <- metagen(TE = lnOR,
                          seTE = SElnOR,
                          studlab = study,
                          data = abs_d_full_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
abs_egger_6m <- metabias(abs_metagen_6m, method.bias = "Egger", k.min = 4)

# Print the results of the Egger's test
print(abs_egger_6m)
```

```
## Linear regression test of funnel plot asymmetry
##
## Test result: t = -0.15, df = 5, p-value = 0.8850
## Bias estimate: -0.1092 (SE = 0.7175)
##
## Details:
## - multiplicative residual heterogeneity variance (tau^2 = 1.4989)
## - 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(abs_metagen_6m,
       xlab = "Odds ratio",
       studlab = TRUE,
       pch = 17,
       cex = 1)

# Add Egger's test statistic to the plot
abs_egger_text_6m <- paste0("Egger's test: p = ",
```

```
round(abs_egger_6m$p.value, 3))

# Define the actual placement of the statistics on the funnel plot
mtext(abs_egger_text_6m, side=3, line=0.5, adj=0, cex=0.9, col="black")
```

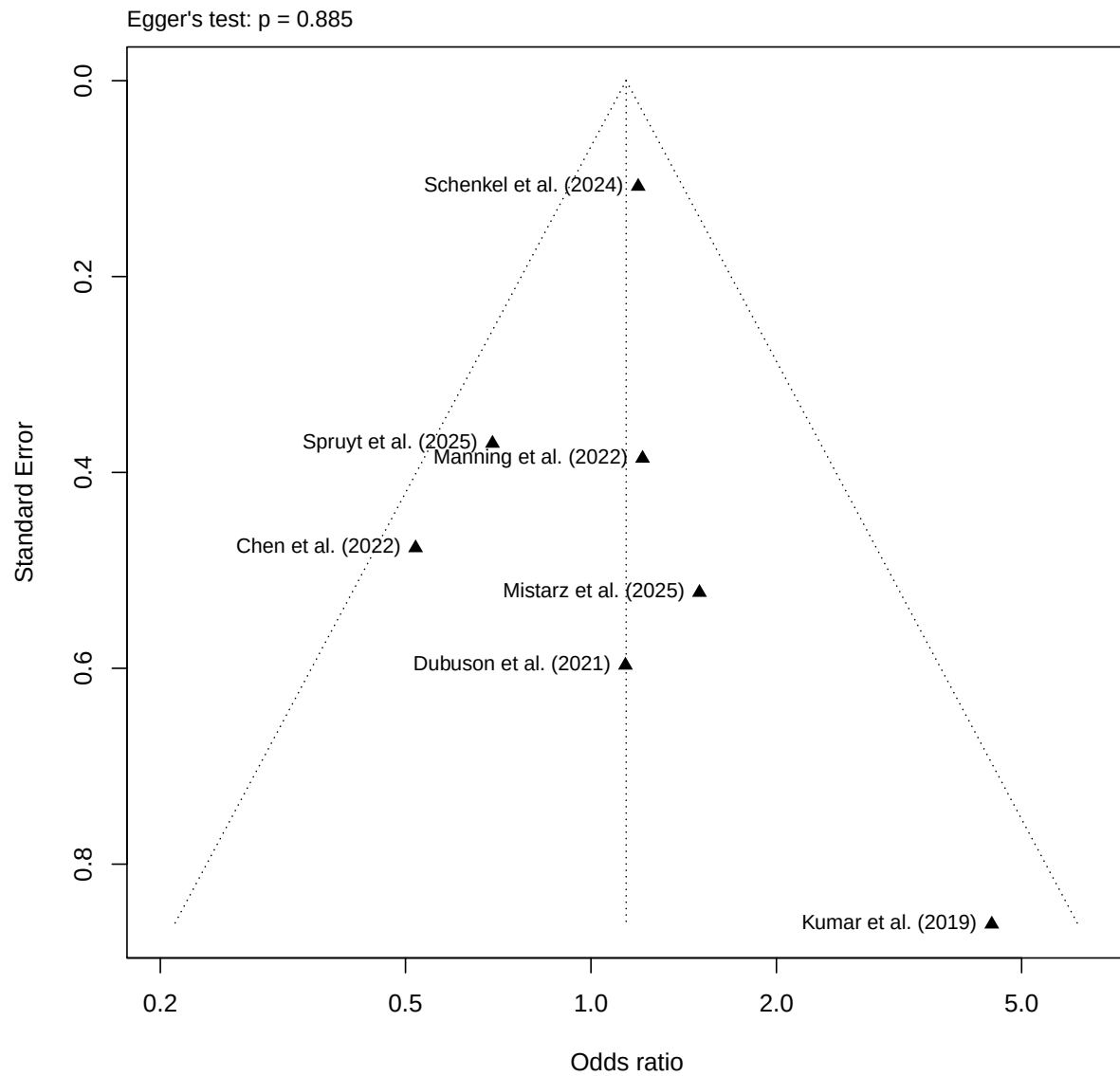


Figure 11: Funnel plot for studies with intermediate-term effects on abstinence

3.2 Intermediate-term alcohol reduction

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

3.2.1 Calculation of effect sizes for intermediate-term continuous variables

We want to follow the procedure of as above 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$time_point,
  system = red_d_6m$system,
  intervention = red_d_6m$intervention,
  total_n = red_d_6m$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$e_reduc_sd,
  # Adding calculated SMDs to the data frame
  SMD = red_metacont_6m$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
)
```

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_6m$study <- ifelse(red_d_full_6m$study %in% names(study_labels),
                             study_labels[red_d_full_6m$study],
                             red_d_full_6m$study)
```

We want to have the author labels 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
red_d_full_6m <- red_d_full_6m %>%
  arrange(study)
```

3.2.2 Meta-analysis of intermediate-term effects on alcohol reduction

Now with the newly defined subgroup labels, we can conduct the meta-analysis using metagen().

```
# Run the meta-analysis stratified by cognitive systems and using the new labels
red_metagen_6m <- metagen(TE = SMD,
                          seTE = SE,
                          studlab = study,
                          data = red_d_full_6m,
                          sm = "SMD",
                          method.tau = "REML",
                          common = FALSE,
                          random = TRUE,
                          method.random.ci = "classic")

# Print the results of the meta-analysis
print(red_metagen_6m)
```

```
## Number of studies: k = 3
##
##               SMD              95% CI      z p-value
## Random effects model 0.1262 [-0.0699; 0.3224] 1.26  0.2072
##
## Quantifying heterogeneity (with 95% CIs):
## tau^2 = 0 [0.0000; 1.4899]; tau = 0 [0.0000; 1.2206]
## I^2 = 0.0% [0.0%; 89.6%]; H = 1.00 [1.00; 3.10]
##
## Test of heterogeneity:
##      Q d.f. p-value
##  1.07   2  0.5868
##
## 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
```

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(red_metagen_6m,
       prediction = TRUE,
```

```

layout = "BMJ",
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_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"
)

```

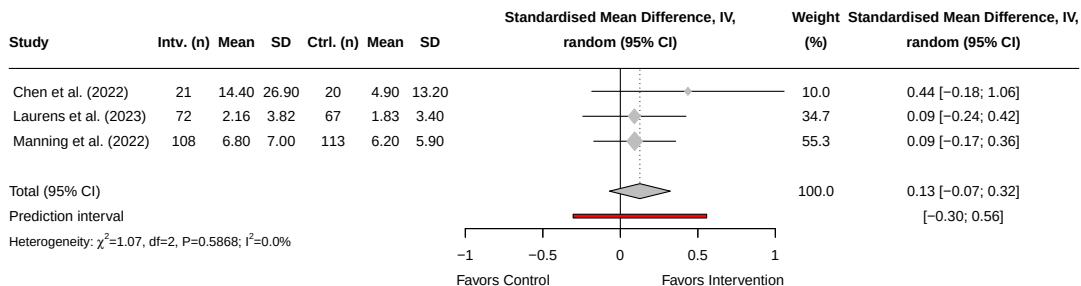


Figure 12: Forest plot of intermediate-term effects on alcohol reduction

3.2.3 Publication bias for intermediate effects on alcohol reduction

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
red_egger_6m <- metabias(red_metagen_6m, method.bias = "Egger", k.min = 2)

```

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_metagen_6m,
       xlab = "Standardised Mean Difference",

```

```

studlab = TRUE,
pch = 17,
cex = 0.5)

# Add Egger's test statistic to the plot
red_egger_text_6m <- paste0("Egger's test: p = ",
                             round(red_egger_6m$p.value, 3))

# Define the actual placement of the statistics on the funnel plot
mtext(red_egger_text_6m, side=3, line=0.5, adj=0, cex=0.9, col="black")

```

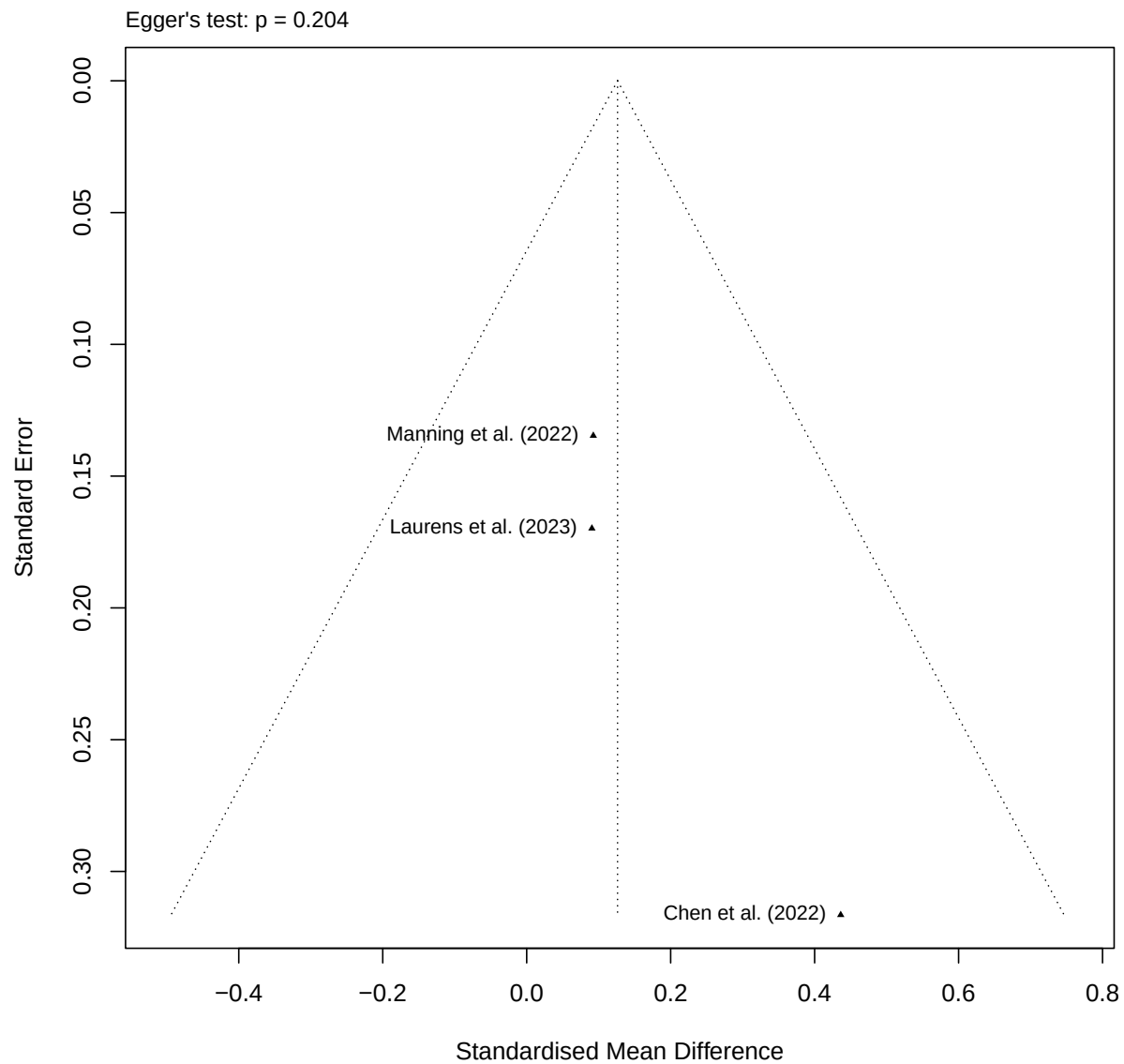


Figure 13: Funnel plot of studies with intermediate-term effects on alcohol reduction

3.3 Intermediate-term craving

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

3.3.1 Calculation of effect sizes for intermediate-term continuous variables

We want to follow the procedure of as above and filter the data by the time point “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 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()
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$time_point,
  system = crav_d_6m$system,
  intervention = crav_d_6m$intervention,
  total_n = crav_d_6m$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$e_crav_sd,
  # Adding calculated SMDs to the data frame
  SMD = crav_metacont_6m$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
)

```

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_6m$study <- ifelse(crav_d_full_6m$study %in% names(study_labels),
                              study_labels[crav_d_full_6m$study],
                              crav_d_full_6m$study)

```

We want to have the author labels 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
crav_d_full_6m <- crav_d_full_6m %>%
  arrange(study)

```

3.2.2 Meta-analysis of intermediate-term effects on alcohol reduction

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 cognitive systems and using the new labels
crav_metagen_6m <- metagen(TE = SMD,
                           seTE = SE,
                           studlab = study,
                           data = crav_d_full_6m,
                           sm = "SMD",
                           method.tau = "REML",
                           common = FALSE,
                           random = TRUE,
                           method.random.ci = "classic")

# Print the results of the meta-analysis
print(crav_metagen_6m)

```

```

## Number of studies: k = 2
##
##              SMD              95% CI      z p-value
## Random effects model -0.8804 [-2.8395; 1.0786] -0.88  0.3784
##
## Quantifying heterogeneity (with 95% CIs):
## tau^2 = 1.9471 [0.3464; >100.0000]; tau = 1.3954 [0.5886; >10.0000]
## I^2 = 97.4% [93.6%; 99.0%]; H = 6.24 [3.94; 9.87]
##
## Test of heterogeneity:
##      Q d.f.  p-value
## 38.90    1 < 0.0001
##
## 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
```

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(crav_metagen_6m,
       prediction = TRUE,
       layout = "BMJ",
       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"
       )
```

3.2.3 Publication bias for intermediate effects on alcohol reduction

As we only identified two studies for the intermediate-term effects on craving, a visual inspection for asymmetry will not be meaningful nor will there be sufficient power for an Egger's test.

4. Main analyses on long-term outcomes

The third and last part of the main analyses will be conducted on long-term effects on outcomes 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 Long-term abstinence

First we conduct the analyses on our primary outcome of abstinence, and we calculate the effect sizes for the long-term time point.

4.1.1 Calculation of effect sizes for long-term continuous variables

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

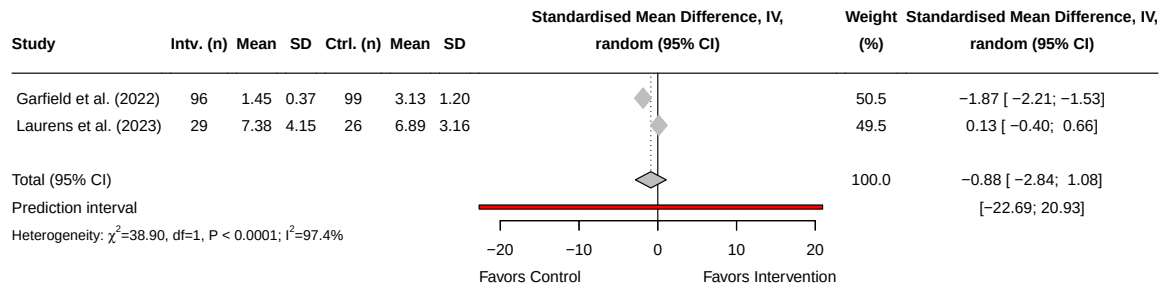


Figure 14: Forest plot of intermediate-term effects on craving

```
# Filter 12 month data first and store it as a new subset
abs_d_12m <- abs_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()
abs_metabin_12m <- metabin(
  event.e = e_abs,
  n.e = e_n,
  event.c = c_abs,
  n.c = c_n,
  studlab = study,
  data = abs_d_12m,
  sm = "OR",
  method.tau = "REML",
  method.random.ci = "classic",
  random = TRUE,
  incr = 0.5,
  allstudies = TRUE
)
```

After we have calculated the $\ln ORs$, we can create a new data frame and store these values alongside with the other relevant variables (e.g., time point, number of abstinent, 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
abs_d_full_12m <- data.frame(
  row_id = abs_d_12m$row_id,
```

```

study = abs_metabin_12m$studlab,
time_point = abs_d_12m$time_point,
system = abs_d_12m$system,
intervention = abs_d_12m$intervention,
total_n = abs_d_12m$e_n + abs_d_12m$c_n,
c_n = abs_d_12m$c_n,
c_abs = abs_d_12m$c_abs,
c_non_abs = abs_d_12m$c_n - abs_d_12m$c_abs,
e_n = abs_d_12m$e_n,
e_abs = abs_d_12m$e_abs,
e_non_abs = abs_d_12m$e_n - abs_d_12m$e_abs,
# Extracting the lnORs from the metabin results
lnOR = abs_metabin_12m$TE,
SElnOR = abs_metabin_12m$seTE,
lnLCI = abs_metabin_12m$lower,
lnUCI = abs_metabin_12m$upper,
# Calculating the ORs from the metabin results
OR = exp(abs_metabin_12m$TE),
OrSE = exp(abs_metabin_12m$TE) * abs_metabin_12m$seTE,
OrLCI = exp(abs_metabin_12m$lower),
OrUCI = exp(abs_metabin_12m$upper),
# Calculating the SMD, SE and 95% CI using the Chinn-function
SMD = abs_metabin_12m$TE * chinn_lnor_to_smd,
SE = abs_metabin_12m$seTE * chinn_lnor_to_smd,
LCI = abs_metabin_12m$lower * chinn_lnor_to_smd,
UCI = abs_metabin_12m$upper * chinn_lnor_to_smd,
# Adding the z- and p-values
`z-value` = abs_metabin_12m$TE / abs_metabin_12m$seTE,
`p-value` = 2 * (1 - pnorm(abs(abs_metabin_12m$TE / abs_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
)

```

We can rename the subgroups for later use for data export.

```

# Assigning new variable names to the subgroups
abs_d_full_12m <- abs_d_full_12m %>%
  dplyr::rename(`Cognitive system` = system) %>%
  dplyr::mutate( # Changing the order of the two subgroups
    `Cognitive system` = factor(`Cognitive system`,
                                levels = c("Implicit", "Explicit"))
  )

```

First we need to assign study labels to the data frame for 12 months.

```

# Assign the new study labels to the combined data frame
abs_d_full_12m <- abs_d_full_12m %>%
  mutate(
    study = ifelse(
      study %in% names(study_labels),
      study_labels[study],
      study
    )
  )

```

We want to have the author labels 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
abs_d_full_12m <- abs_d_full_12m %>%
  arrange(study)
```

We can now calculate the baseline control risk for the long-term effects by only including studies with categorical outcomes.

```
# Abstinence rate risk difference across all studies on short-term effects
p_c_abs_12m <- with(abs_d_full_12m, sum(c_abs) / sum(c_n))
```

4.1.2 Meta-analysis of long-term effects on abstinence

Now with the newly defined subgroup labels, we can conduct the meta-analysis using metagen().

```
# Run the meta-analysis stratified by cognitive systems and using the new labels
abs_metagen_12m <- metagen(TE = lnOR,
  seTE = SElnOR,
  studlab = study,
  data = abs_d_full_12m,
  sm = "OR",
  method.tau = "REML",
  common = FALSE,
  random = TRUE,
  method.random.ci = "classic")

# Print the results of the meta-analysis
print(abs_metagen_12m)
```

```
## Number of studies: k = 11
##
##              OR          95% CI      z p-value
## Random effects model 1.2167 [1.0397; 1.4239] 2.44 0.0145
##
## Quantifying heterogeneity (with 95% CIs):
## tau^2 = 0.0178 [0.0000; 0.3260]; tau = 0.1334 [0.0000; 0.5709]
## I^2 = 30.8% [0.0%; 66.0%]; H = 1.20 [1.00; 1.71]
##
## Test of heterogeneity:
##      Q d.f. p-value
## 14.46  10 0.1532
##
## 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
abs_zp_overall_12m <- data.frame(
  lnOR = abs_metagen_12m$TE.random,
  lnOR_se = abs_metagen_12m$seTE.random,
```

```

NNT = round(nnt(exp(abs_metagen_12m$TE.random), p_c_abs_12m)),
z_value = abs_metagen_12m$statistic.random,
p_value = abs_metagen_12m$pval.random
)

# Extract the test statistics for the subgroup analysis and create a data frame
abs_zp_subgroup_12m <- data.frame(
  subgroup = abs_metagen_12m$bylevs,
  lnOR = abs_metagen_12m$TE.random.w,
  lnOR_se = abs_metagen_12m$seTE.random.w,
  NNT = round(nnt(abs_metagen_12m$TE.random.w, p_c_abs_12m)),
  z_value = abs_metagen_12m$statistic.random.w,
  p_value = abs_metagen_12m$pval.random.w
)

```

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(abs_metagen_12m,
  prediction = TRUE,
  layout = "BMJ",
  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,
  xlim = c(1/15, 15),
  at = c(0.07, 0.2, 0.5, 1, 2, 5, 15),
  leftcols = c("studlab", "e_abs", "e_n", "c_abs", "c_n"),
  leftlabs = c("Study", "Intv. (events)", "Total (n)", "Ctrl. (events)", "Total (n)"),
  rightcols = c("w.random", "effect", "ci"),
  rightlabs = c("Weight (%)", "OR", "95% CI")
)

footer_over_abs_12m <- paste0(
  "Overall effect: ",
  "z = ", round(abs_zp_overall_12m$z_value, 2),
  ", p = ", formatC(abs_zp_overall_12m$p_value, digits = 2),
  ", NNT = ", abs_zp_overall_12m$NNT
)

# Adjust the footer for the overall effect to the bottom left
grid.text(footer_over_abs_12m, x = 0.017, y = 0.06,
  just = "left", gp = gpar(fontsize = 10))

```

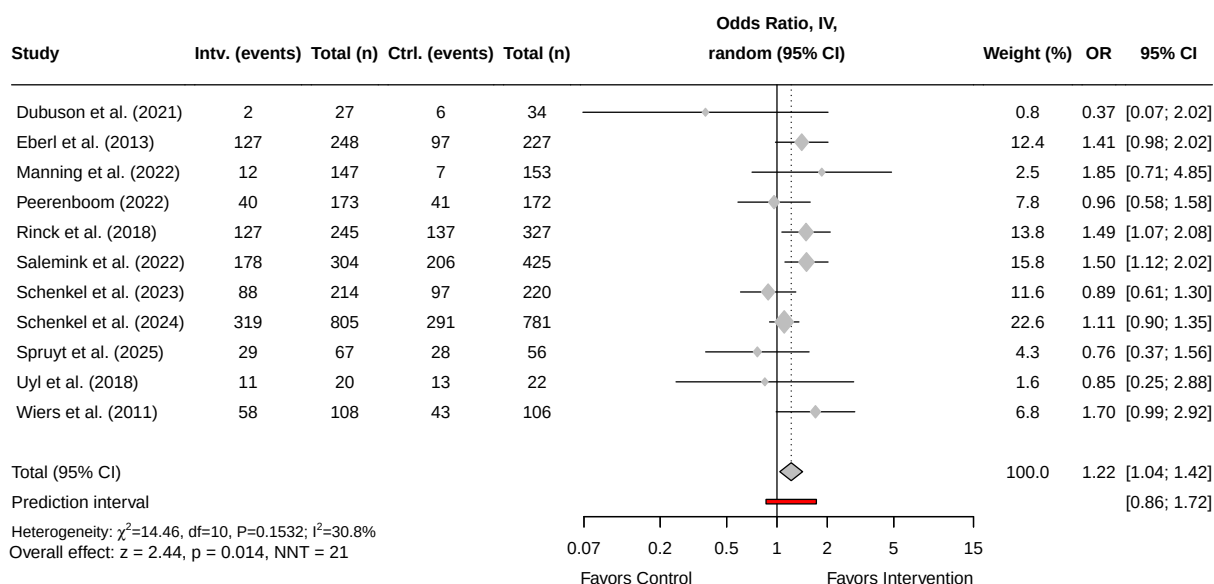


Figure 15: Forest plot for the long-term effects on abstinence

4.1.3 Publication bias for long-term effects on abstinence

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
abs_egger_12m <- metabias(abs_metagen_12m, method.bias = "Egger", k.min = 8)
```

```
# Print the results of the Egger's test
print(abs_egger_12m)
```

```
## Linear regression test of funnel plot asymmetry
##
## Test result: t = -0.60, df = 9, p-value = 0.5650
## Bias estimate: -0.4617 (SE = 0.7729)
##
## Details:
## - multiplicative residual heterogeneity variance (tau^2 = 1.5449)
## - 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(abs_metagen_12m,
  xlab = "Odds Ratio",
  studlab = TRUE,
  pch = 17,
```

```

cex = 0.5)

# Add Egger's test statistic to the plot
abs_egger_text_12m <- paste0("Egger's test: p = ",
                             round(abs_egger_12m$p.value, 3))

# Define the actual placement of the statistics on the funnel plot
mtext(abs_egger_text_12m, side=3, line=0.5, adj=0, cex=0.9, col="black")

```

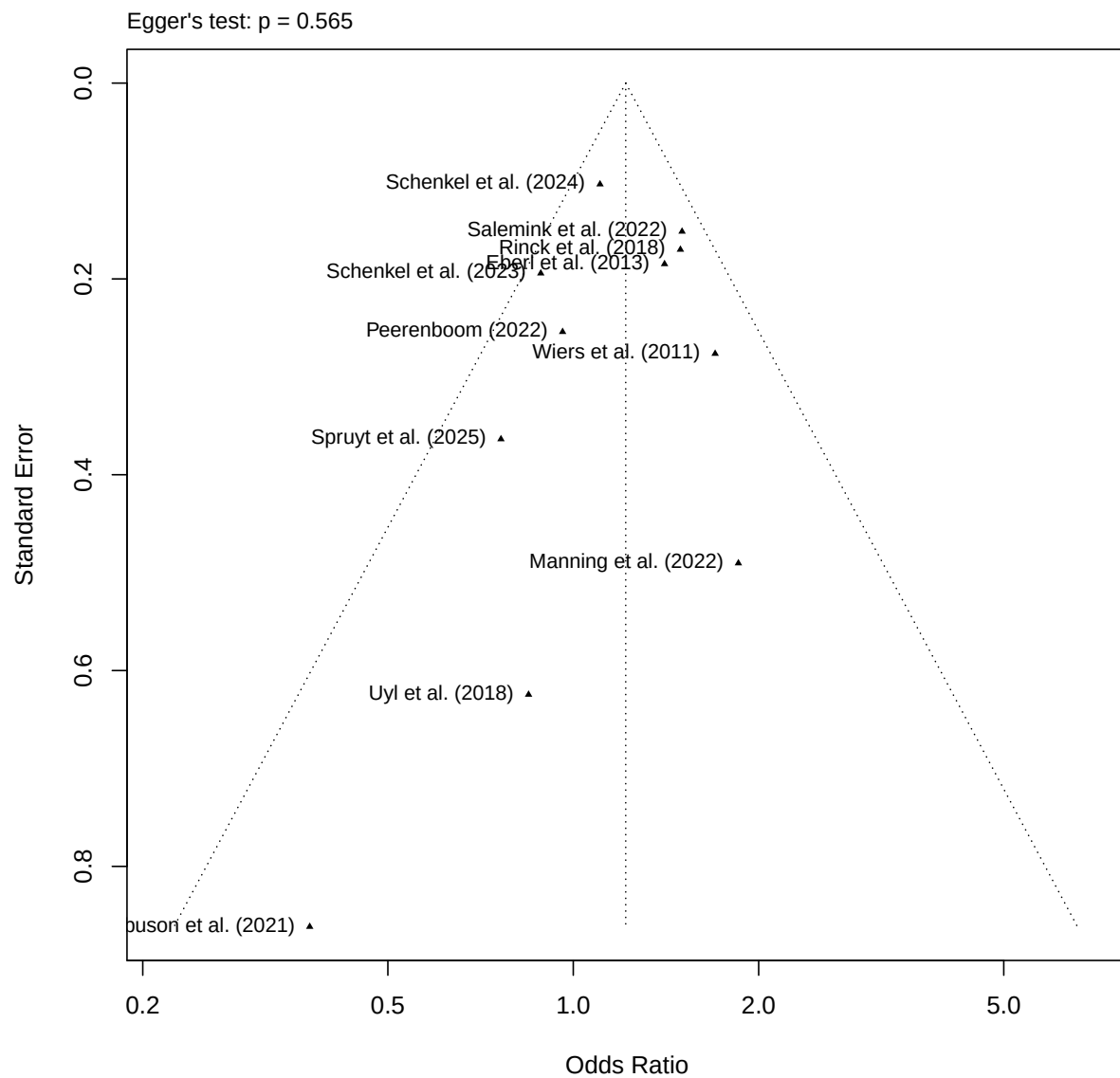


Figure 16: Funnel plot of studies with long-term effects on abstinence

4.2 Long-term effects on alcohol reduction

4.2.1 Calculation of effect sizes for long-term continuous 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$time_point,
  system = red_d_12m$system,
  intervention = red_d_12m$intervention,
  total_n = red_d_12m$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$e_reduc_sd,
  # Adding calculated SMDs to the data frame
  SMD = red_metacont_12m$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
)
```

```
)
```

4.2.2 Meta-analysis of long-term effects on alcohol reduction

As we only identified one study examining long-term effects on alcohol reduction, a meta-analysis could not be performed.

4.2.3 Publication bias for long-term effects on alcohol reduction

An assessment of publication bias was not possible due to insufficient studies.

4.3 Long-term effects on craving

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

4.3.1 Calculation of effect sizes for long-term continuous variables

We repeat the methodology used previously and start with filtering the data for 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$time_point,
  system = crav_d_12m$system,
  intervention = crav_d_12m$intervention,
  total_n = crav_d_12m$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$e_crav_sd,
# Adding calculated SMDs to the data frame
SMD = crav_metacont_12m$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
)

```

4.3.2 Meta-analysis of long-term effects on craving

As only one study was identified on long-term effects on craving, no meta-analysis could be performed.

4.3.3 Publication bias for intermediate effects on craving

An assessment of publication bias was not possible due to insufficient studies.

5. Exploratory intervention-specific analyses

To explore the short-term effects on abstinence by each intervention type, we use the assigned new variable names to the subgroups to create a visually more appealing forest plot. We use the labels for our intervention types: AtBM (Attentional bias modification), ApBM (Approach bias modification), ItBM (Inhibitory bias modification), ExRT (Executive function retraining), and WmRT (Working memory retraining).

```

# Assigning new variable names to the subgroups
abs_d_full_3m <- abs_d_full_3m %>%
  rename(Intervention = intervention) %>%
  mutate(
    Intervention = factor(
      Intervention,
      levels = c("AtBM", "ItBM", "ApBM", "ExRT", "WmRT")
    )
  )

```

5.1 Short-term effects on abstinence by intervention

We conduct the exploratory analyses first on the short-term abstinence, which included the most studies.

5.1.1 Meta-analysis on abstinence

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

With the new subgroup names, we can pool the effects across studies and stratify by the cognitive systems targeted by the intervention 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 cognitive systems and using the new labels
```

```
abs_exp_metagen_3m <- metagen(TE = lnOR,
  seTE = SElnOR,
  studlab = study,
  data = abs_d_full_3m,
  subgroup = Intervention,
  sm = "OR",
  method.tau = "REML",
  common = FALSE,
  random = TRUE,
  method.random.ci = "classic")
```

```
# Print the results of the meta-analysis
```

```
print(abs_exp_metagen_3m)
```

```
## Number of studies: k = 13
##
##              OR          95% CI    z p-value
## Random effects model 1.2430 [1.0789; 1.4321] 3.01  0.0026
##
## Quantifying heterogeneity (with 95% CIs):
## tau^2 < 0.0001 [0.0000; 0.8164]; tau = 0.0016 [0.0000; 0.9035]
## I^2 = 32.6% [0.0%; 65.2%]; H = 1.22 [1.00; 1.69]
##
## Test of heterogeneity:
##      Q d.f. p-value
## 17.80  12  0.1218
##
## Results for subgroups (random effects model):
##      k      OR          95% CI  tau^2    tau    Q    I^2
## Intervention = AtBM  2 0.6778 [0.2464; 1.8643]    0    0 0.05  0.0%
## Intervention = ItBM  4 1.2208 [0.7529; 1.9798] 0.0972 0.3117 5.33 43.7%
## Intervention = ApBM  5 1.2548 [1.0632; 1.4808]    0    0 2.25  0.0%
## Intervention = ExRT  1 7.1111 [1.9858; 25.4649]   --    -- 0.00   --
## Intervention = WmRT  1 2.0000 [0.7717; 5.1837]   --    -- 0.00   --
##
## Test for subgroup differences (random effects model):
##      Q d.f. p-value
## Between groups 9.39  4  0.0520
##
## 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
```

```
abs_exp_zp_overall_3m <- data.frame(
```

```

lnOR = abs_exp_metagen_3m$TE.random,
lnOR_se = abs_exp_metagen_3m$seTE.random,
NNT = round(nnt(exp(abs_exp_metagen_3m$TE.random), p_c_abs_3m)),
z_value = abs_exp_metagen_3m$statistic.random,
p_value = abs_exp_metagen_3m$pval.random
)

# Extract the test statistics for the subgroup analysis and create a data frame
abs_exp_zp_subgroup_3m <- data.frame(
  subgroup = abs_exp_metagen_3m$bylevs,
  lnOR = abs_exp_metagen_3m$TE.random.w,
  lnOR_se = abs_exp_metagen_3m$seTE.random.w,
  NNT = round(nnt(exp(abs_exp_metagen_3m$TE.random.w), p_c_abs_3m)),
  z_value = abs_exp_metagen_3m$statistic.random.w,
  p_value = abs_exp_metagen_3m$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
abs_exp_subsplit_3m <- split(abs_exp_zp_subgroup_3m, abs_exp_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(abs_exp_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",
  # Right side: default BMJ columns
  rightcols = c("w.random", "effect", "ci"),
  rightlabs = c("Weight (%)", "OR", "95% CI")
)

```

```

# Add a footer for the test statistics of the effects for AtBM
footer_atbm_abs_3m <- paste0(
  "Subgroup effect ",
  paste0(
    ": z = ", round(abs_exp_subsplit_3m$AtBM$z_value, 2),
    ", p = ", formatC(abs_exp_subsplit_3m$AtBM$p_value, digits = 2),
    ", NNH = ", abs(abs_exp_subsplit_3m$AtBM$NNT) # Write NNH instead, as it is negative
  )
)

# Add a footer for the test statistics of the effects for ItBM
footer_itbm_abs_3m <- paste0(
  "Subgroup effect ",
  paste0(
    ": z = ", round(abs_exp_subsplit_3m$ItBM$z_value, 2),
    ", p = ", formatC(abs_exp_subsplit_3m$ItBM$p_value, digits = 2),
    ", NNT = ", abs_exp_subsplit_3m$ItBM$NNT
  )
)

# Add a footer for the test statistics of the effects for ApBM
footer_apbm_abs_3m <- paste0(
  "Subgroup effect ",
  paste0(
    ": z = ", round(abs_exp_subsplit_3m$ApBM$z_value, 2),
    ", p = ", formatC(abs_exp_subsplit_3m$ApBM$p_value, digits = 2),
    ", NNT = ", abs_exp_subsplit_3m$ApBM$NNT
  )
)

# Add a footer for the test statistics of the overall effects
footer_over_abs_3m <- paste0(
  "Overall effect ",
  paste0(
    ": z = ", round(abs_exp_zp_overall_3m$z_value, 2),
    ", p = ", formatC(abs_exp_zp_overall_3m$p_value, digits = 2),
    ", NNT = ", abs_exp_zp_overall_3m$NNT
  )
)

# Adjust the footer for the AtBM effect to the bottom left
grid.text(footer_atbm_abs_3m, x = 0.076, y = 0.73,
  just = "left", gp = gpar(fontsize = 10))

# Adjust the footer for the ItBM effect to the bottom left
grid.text(footer_itbm_abs_3m, x = 0.076, y = 0.51,
  just = "left", gp = gpar(fontsize = 10))

# Adjust the footer for the ApBM effect to the bottom left
grid.text(footer_apbm_abs_3m, x = 0.076, y = 0.27,
  just = "left", gp = gpar(fontsize = 10))

# Adjust the footer for the overall effect to the bottom left

```

```
grid.text(footer_over_abs_3m, x = 0.076, y = 0.035,
just = "left", gp = gpar(fontsize = 10))
```

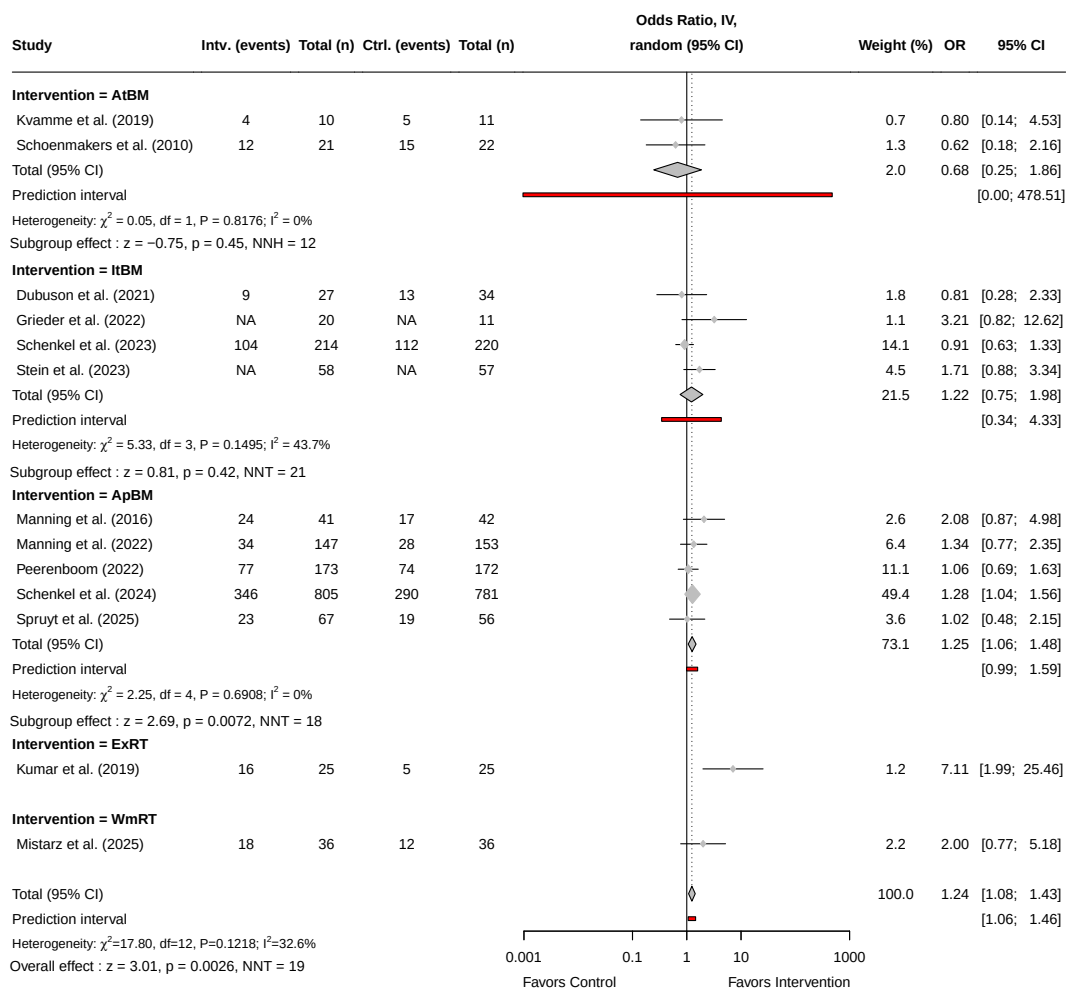


Figure 17: Forest plot of the short-term effects on abstinence by individual intervention types

5.2 Short-term effects on alcohol reduction by intervention

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 from the main analyses.

5.2.1 Meta-analysis on alcohol reduction

Not possible to pool effects for other interventions than ApBM, thus the results are the same as for the main analyses.

5.3 Short-term effects on alcohol craving by intervention

To explore the short-term effects on craving by each intervention type, we use the assigned new variable names to the subgroups to create a visually more appealing forest plot. We use the labels for our intervention types: AtBM (Attentional bias modification), ApBM (Approach bias modification), ItBM (Inhibitory bias modification), ExRT (Executive function retraining), and WmRT (Working memory retraining).

```
# Assigning new variable names to the subgroups
crav_d_full_3m <- crav_d_full_3m %>%
  rename(Intervention = intervention) %>%
  mutate(
    Intervention = factor(
      Intervention,
      levels = c("AtBM", "ItBM", "ApBM", "ExRT", "WmRT")
    )
  )
```

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

With the new subgroup names, we can pool the effects across studies and stratify by the cognitive systems targeted by the intervention used in each study.

This is achieved by using `metagen()`, where we define our treatment effect (TE) as the SMD and corresponding standard error of the treatment effect (seTE) as the SE of the SMD. We apply a random effects model using REML and use a classic z-test.

```
# Run the meta-analysis stratified by cognitive systems and using the new labels
crav_exp_metagen_3m <- metagen(TE = SMD,
  seTE = SE,
  studlab = study,
  data = crav_d_full_3m,
  subgroup = Intervention,
  sm = "SMD",
  method.tau = "REML",
  common = FALSE,
  random = TRUE,
  method.random.ci = "classic")

# Print the results of the meta-analysis
print(crav_exp_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    I^2
```

```
## Intervention = AtBM    3 -0.0164 [-0.4063; 0.3735] <0.0001 0.0021  2.58 22.4%
## Intervention = ItBM    2 -0.1558 [-0.6193; 0.3076]  0.0573 0.2394  2.03 50.7%
## Intervention = ApBM    4 -0.1015 [-0.6085; 0.4054]  0.2291 0.4787 21.10 85.8%
##
## 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_exp_zp_overall_3m <- data.frame(
  SMD = crav_exp_metagen_3m$TE.random,
  SE = crav_exp_metagen_3m$seTE.random,
  z_value = crav_exp_metagen_3m$statistic.random,
  p_value = crav_exp_metagen_3m$pval.random
)

# Extract the test statistics for the subgroup analysis and create a data frame
crav_exp_zp_subgroup_3m <- data.frame(
  subgroup = crav_exp_metagen_3m$bylevs,
  SMD = crav_exp_metagen_3m$TE.random.w,
  SE = crav_exp_metagen_3m$seTE.random.w,
  z_value = crav_exp_metagen_3m$statistic.random.w,
  p_value = crav_exp_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_exp_subsplit_3m <- split(crav_exp_zp_subgroup_3m, crav_exp_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_exp_metagen_3m,
  prediction = TRUE,
  prediction.subgroup = TRUE,
  layout = "BMJ",
  overall = TRUE,
  subgroup = 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"
)

```

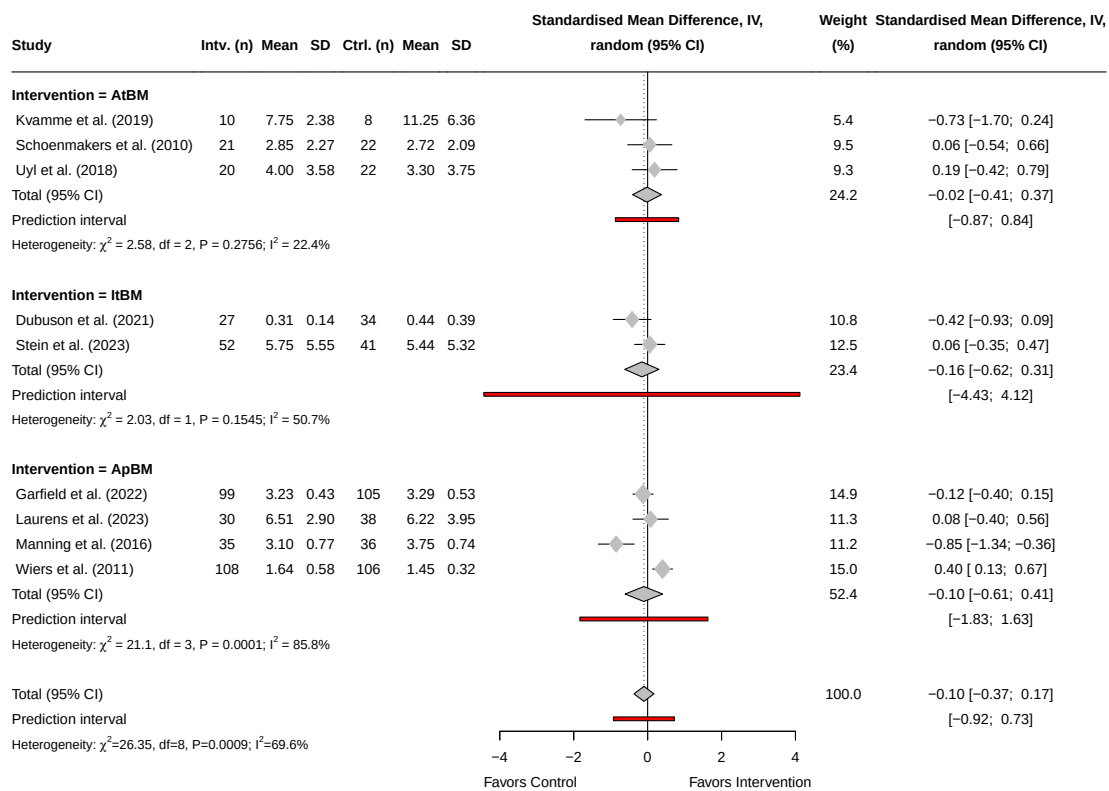


Figure 18: Forest plot of the short-term effects on craving by individual intervention type

5.3.1 Meta-analysis on alcohol craving

Insufficient data for analysis for intervention types. For each subgroup in the final forest plot, we can create a split of the data frame.

5.4 Intermediate-term effects on abstinence by intervention

Now we can look at the intermediate-term effects on abstinence by each intervention type. First we define the intervention types and its levels.

```

# Assigning new variable names to the subgroups
abs_d_full_6m <- abs_d_full_6m %>%
  rename(Intervention = intervention) %>%
  mutate(
    Intervention = factor(
      Intervention,
      levels = c("AtBM", "ItBM", "ApBM", "ExRT", "WmRT")
    )
  )

```

5.4.1 Meta-analysis on abstinence

Then we run a meta-analysis using this new subset of our abstinence data for 6 months.

```

# Run the meta-analysis stratified by cognitive systems and using the new labels
abs_exp_metagen_6m <- metagen(TE = lnOR,
  seTE = SElnOR,
  studlab = study,
  data = abs_d_full_6m,
  subgroup = Intervention,
  sm = "OR",
  method.tau = "REML",
  common = FALSE,
  random = TRUE,
  method.random.ci = "classic")

# Print the results of the meta-analysis
print(abs_exp_metagen_6m)

```

```

## Number of studies: k = 7
##
##              OR           95% CI    z p-value
## Random effects model 1.1406 [0.9478; 1.3726] 1.39 0.1639
##
## Quantifying heterogeneity (with 95% CIs):
## tau^2 < 0.0001 [0.0000; 1.7423]; tau = 0.0007 [0.0000; 1.3200]
## I^2 = 20.3% [0.0%; 63.8%]; H = 1.12 [1.00; 1.66]
##
## Test of heterogeneity:
##      Q d.f. p-value
## 7.53    6 0.2746
##
## Results for subgroups (random effects model):
##      k      OR           95% CI  tau^2    tau    Q    I^2
## Intervention = ItBM  1 1.1375 [0.3531; 3.6648]    --    --  0.00    --
## Intervention = ApBM  4 0.9803 [0.6825; 1.4081] 0.0528 0.2299 4.66 35.6%
## Intervention = ExRT  1 4.4722 [0.8267; 24.1930]    --    --  0.00    --
## Intervention = WmRT  1 1.5000 [0.5385; 4.1779]    --    --  0.00    --
##
## Test for subgroup differences (random effects model):
##      Q d.f. p-value
## Between groups 3.39    3 0.3348
##
## 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
```

We can now calculate the baseline control risk for the intermediate-term effects by only including studies with categorical outcomes.

```
# Abstinence rate risk difference across all studies on short-term effects
p_c_exp_abs_6m <- with(abs_d_6m, sum(c_abs) / sum(c_n))
```

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
abs_exp_zp_overall_6m <- data.frame(
  lnOR = abs_exp_metagen_6m$TE.random,
  lnOR_se = abs_exp_metagen_6m$seTE.random,
  NNT = round(nnt(exp(abs_exp_metagen_6m$TE.random), p_c_exp_abs_6m)),
  z_value = abs_exp_metagen_6m$statistic.random,
  p_value = abs_exp_metagen_6m$pval.random
)

# Extract the test statistics for the subgroup analysis and create a data frame
abs_exp_zp_subgroup_6m <- data.frame(
  subgroup = abs_exp_metagen_6m$bylevs,
  lnOR = abs_exp_metagen_6m$TE.random.w,
  lnOR_se = abs_exp_metagen_6m$seTE.random.w,
  NNT = round(nnt(exp(abs_exp_metagen_6m$TE.random.w), p_c_exp_abs_6m)),
  z_value = abs_exp_metagen_6m$statistic.random.w,
  p_value = abs_exp_metagen_6m$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
abs_exp_subsplit_6m <- split(abs_exp_zp_subgroup_6m, abs_exp_zp_subgroup_6m$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(abs_exp_metagen_6m,
  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",
rightcols = c("w.random", "effect", "ci"),
rightlabs = c("Weight (%)", "OR", "95% CI")
)

# Add a footer for the test statistics of the effects for ApBM
footer_apbm_abs_6m <- paste0(
  "Subgroup effect ",
  paste0(
    ": z = ", round(abs_exp_subsplit_6m$ApBM$z_value, 2),
    ", p = ", formatC(abs_exp_subsplit_6m$ApBM$p_value, digits = 2),
    ", NNH > 100 "
  )
)

# Add a footer for the test statistics of the overall effects
footer_over_abs_6m <- paste0(
  "Overall effect ",
  paste0(
    ": z = ", round(abs_exp_zp_overall_6m$z_value, 2),
    ", p = ", formatC(abs_exp_zp_overall_6m$p_value, digits = 2),
    ", NNT = ", abs_exp_zp_overall_6m$NNT
  )
)

# Adjust the footer for the ApBM effect to the bottom left
grid.text(footer_apbm_abs_6m, x = 0.094, y = 0.43,
          just = "left", gp = gpar(fontsize = 10))

# Adjust the footer for the overall effect to the bottom left
grid.text(footer_over_abs_6m, x = 0.094, y = 0.12,
          just = "left", gp = gpar(fontsize = 10))

```

5.5 Intermediate-term effects on alcohol reduction by intervention

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 from the main analyses on the intermediate-term outcome.

5.5.1 Meta-analysis on alcohol reduction

Not possible to pool effects for other interventions than ApBM.

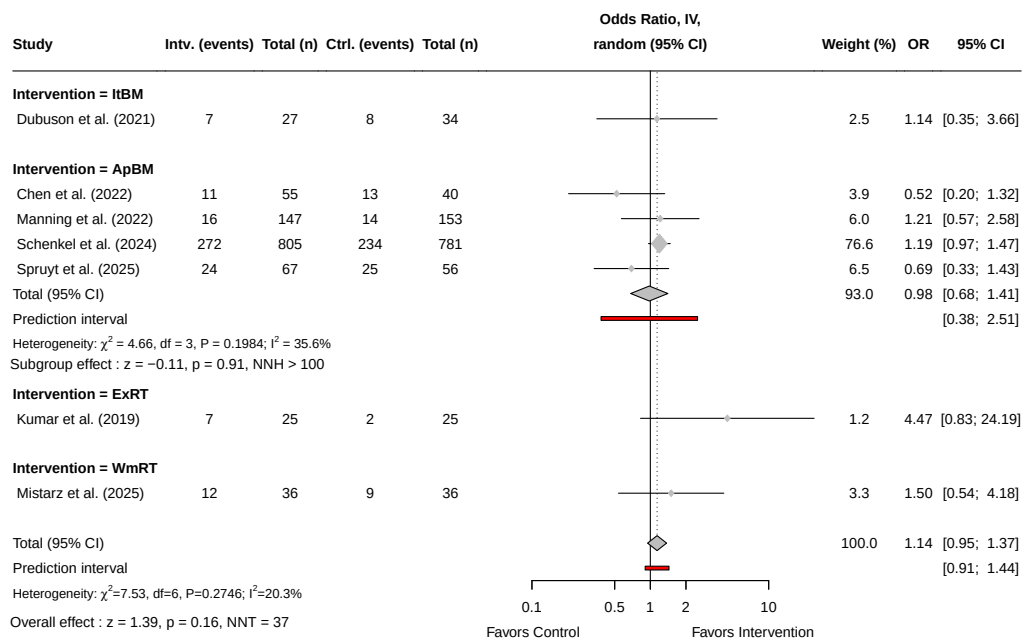


Figure 19: Forest plot of the intermediate-term effects on abstinence by individual intervention types

5.6 Intermediate-term effects on craving by intervention

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 from the main analyses on the intermediate-term outcome.

5.6.1 Meta-analysis on craving

Not possible to pool effects for other interventions than ApBM.

5.7 Long-term effects on abstinence by intervention

Now we can look at the final part of the exploratory analyses, by examining the long-term effects on abstinence by intervention type. First we define the intervention types and its levels.

```
# Assigning new variable names to the subgroups
abs_d_full_12m <- abs_d_full_12m %>%
  rename(Intervention = intervention) %>%
  mutate(
    Intervention = factor(
      Intervention,
      levels = c("AtBM", "ItBM", "ApBM", "ExRT", "WmRT")
    )
  )
```

5.7.1 Meta-analysis on abstinence

Then we run a meta-analysis using this new subset of our abstinence data for 12 months.

```
# Run the meta-analysis stratified by cognitive systems and using the new labels
abs_exp_metagen_12m <- metagen(TE = lnOR,
  seTE = SElnOR,
  studlab = study,
  data = abs_d_full_12m,
  subgroup = Intervention,
  sm = "OR",
  method.tau = "REML",
  common = FALSE,
  random = TRUE,
  method.random.ci = "classic")
```

```
# Print the results of the meta-analysis
print(abs_exp_metagen_12m)
```

```
## Number of studies: k = 11
##
##              OR              95% CI      z p-value
## Random effects model 1.2167 [1.0397; 1.4239] 2.44  0.0145
##
## Quantifying heterogeneity (with 95% CIs):
## tau^2 = 0.0178 [0.0000; 0.3260]; tau = 0.1334 [0.0000; 0.5709]
## I^2 = 30.8% [0.0%; 66.0%]; H = 1.20 [1.00; 1.71]
##
## Test of heterogeneity:
##      Q d.f. p-value
```

```
## 14.46 10 0.1532
##
## Results for subgroups (random effects model):
##           k      OR      95% CI  tau^2    tau    Q    I^2
## Intervention = AtBM  1 0.8462 [0.2488; 2.8778]    --    -- 0.00    --
## Intervention = ItBM  2 0.8494 [0.5859; 1.2314]     0     0 0.96  0.0%
## Intervention = ApBM  8 1.2893 [1.1043; 1.5053] 0.0106 0.1031 9.11 23.2%
##
## Test for subgroup differences (random effects model):
##           Q d.f. p-value
## Between groups 4.46    2 0.1076
##
## 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
```

We can now calculate the baseline control risk for the long-term effects by only including studies with categorical outcomes.

```
# Abstinence rate risk difference across all studies on short-term effects
p_c_exp_abs_12m <- with(abs_d_12m, sum(c_abs) / sum(c_n))
```

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
abs_exp_zp_overall_12m <- data.frame(
  lnOR = abs_exp_metagen_12m$TE.random,
  lnOR_se = abs_exp_metagen_12m$seTE.random,
  NNT = round(nnt(exp(abs_exp_metagen_12m$TE.random), p_c_abs_12m)),
  z_value = abs_exp_metagen_12m$statistic.random,
  p_value = abs_exp_metagen_12m$pval.random
)

# Extract the test statistics for the subgroup analysis and create a data frame
abs_exp_zp_subgroup_12m <- data.frame(
  subgroup = abs_exp_metagen_12m$bylevs,
  lnOR = abs_exp_metagen_12m$TE.random.w,
  lnOR_se = abs_exp_metagen_12m$seTE.random.w,
  NNT = round(nnt(exp(abs_exp_metagen_12m$TE.random.w), p_c_abs_12m)),
  z_value = abs_exp_metagen_12m$statistic.random.w,
  p_value = abs_exp_metagen_12m$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
abs_exp_subsplit_12m <- split(abs_exp_zp_subgroup_12m, abs_exp_zp_subgroup_12m$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(abs_exp_metagen_12m,
```

```

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",
rightcols = c("w.random", "effect", "ci"),
rightlabs = c("Weight (%)", "OR", "95% CI")
)

# Add a footer for the test statistics of the effects for ItBM
footer_itbm_abs_12m <- paste0(
  "Subgroup effect ",
  paste0(
    ": z = ", round(abs_exp_subsplit_12m$ItBM$z_value, 2),
    ", p = ", formatC(abs_exp_subsplit_12m$ItBM$p_value, digits = 2),
    ", NNH = ", abs(abs_exp_subsplit_12m$ItBM$NNT) # Write NNH instead as the number is negative
  )
)

# Add a footer for the test statistics of the effects for ApBM
footer_apbm_abs_12m <- paste0(
  "Subgroup effect ",
  paste0(
    ": z = ", round(abs_exp_subsplit_12m$ApBM$z_value, 2),
    ", p = ", formatC(abs_exp_subsplit_12m$ApBM$p_value, digits = 2),
    ", NNT = ", abs_exp_subsplit_12m$ApBM$NNT
  )
)

# Add a footer for the test statistics of the overall effects
footer_over_abs_12m <- paste0(
  "Overall effect ",
  paste0(
    ": z = ", round(abs_exp_zp_overall_12m$z_value, 2),
    ", p = ", formatC(abs_exp_zp_overall_12m$p_value, digits = 2),

```

```

", NNT = ", abs_exp_zp_overall_12m$NNT
)
)

# Adjust the footer for the ItBM effect to the bottom left
grid.text(footer_itbm_abs_12m, x = 0.098, y = 0.56,
          just = "left", gp = gpar(fontsize = 10))

# Adjust the footer for the ApBM effect to the bottom left
grid.text(footer_apbm_abs_12m, x = 0.098, y = 0.174,
          just = "left", gp = gpar(fontsize = 10))

# Adjust the footer for the overall effect to the bottom left
grid.text(footer_over_abs_12m, x = 0.098, y = 0.05,
          just = "left", gp = gpar(fontsize = 10))

```

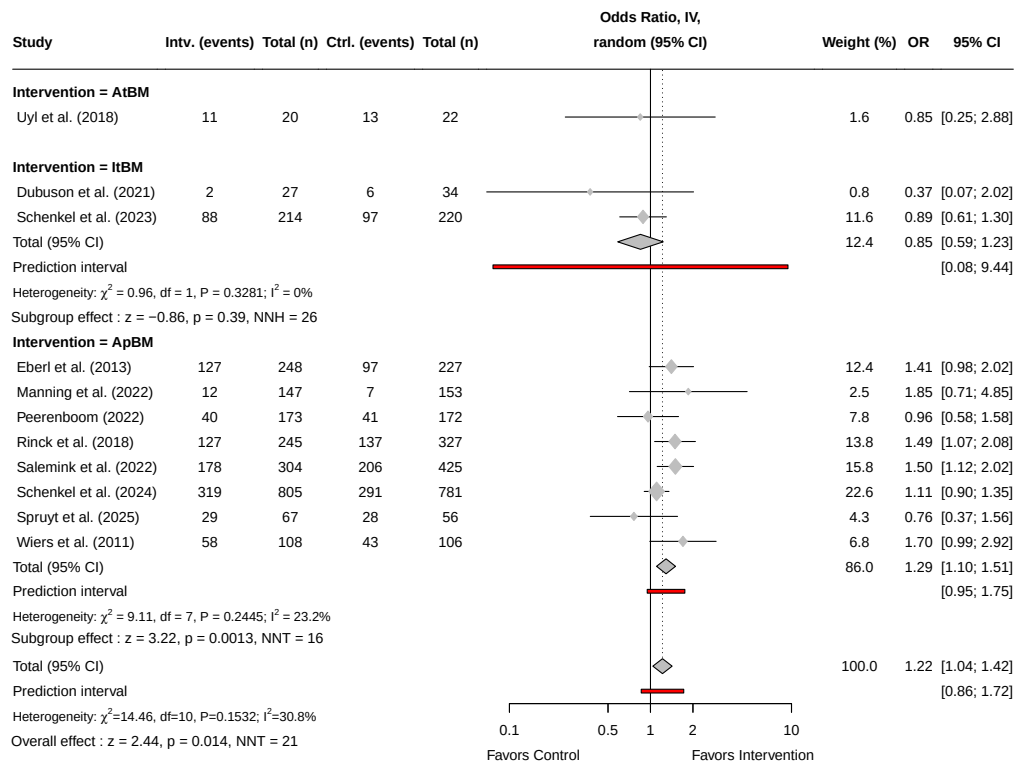


Figure 20: Forest plot of the long-term effects on abstinence by individual intervention type

5.8 Long-term effects on alcohol reduction by intervention

As only one study was identified examining long-term outcomes on alcohol reduction, the exploratory analysis by intervention type could not be performed.

5.8.1 Meta-analysis on alcohol reduction

Meta-analysis was not performed due to insufficient studies.

5.9 Long-term effects on craving by intervention

As only one study was identified examining long-term outcomes on craving, the exploratory analysis by intervention type could not be performed.

5.9.1 Meta-analysis on craving

Meta-analysis was not performed due to insufficient studies.

6. 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.

6.1 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 merge all data frames containing the calculated effect sizes for abstinence, alcohol reduction, and craving into one list of study labels.

```
# Define function to fix the data types for the abs data frames
fix_abs_types <- function(df) {
  df %>%
    mutate(across(
      any_of(c(
        "total_n",
        "c_n", "c_abs", "c_non_abs",
        "e_n", "e_abs", "e_non_abs"
      )),
      ~ suppressWarnings(as.numeric(.))
    ))
}

# Fix data frames before merging
abs_d_full_3m <- fix_abs_types(abs_d_full_3m)
abs_d_full_6m <- fix_abs_types(abs_d_full_6m)
abs_d_full_12m <- fix_abs_types(abs_d_full_12m)

# Combine all data frames
abs_d_full_012m <- bind_rows(
  abs_d_full_3m,
  abs_d_full_6m,
  abs_d_full_12m
) %>%
  arrange(time_point, `Cognitive system`, Intervention, study) %>%
  mutate(row_id = row_number()) %>%
  rename(
    system = `Cognitive system`,
```

```

    intervention = Intervention
  )

# ALCOHOL REDUCTION DATA FRAMES
# Define function to fix the data types for the red data frames
fix_red_types <- function(df) {
  df %>%
    mutate(across(
      any_of(c(
        # counts / sample sizes
        "total_n", "c_n", "e_n",
        # continuous outcome descriptives (if present)
        "c_mean", "e_mean", "c_sd", "e_sd",
        # effect-size columns (if present)
        "SMD", "SE", "LCI", "UCI",
        "lnOR", "SElnOR", "lnLCI", "lnUCI",
        "OR", "OrSE", "OrLCI", "OrUCI",
        "z_value", "p_value"
      )),
      ~ suppressWarnings(as.numeric(.))
    ))
}

# Fix data frames before merging
red_d_full_3m <- fix_red_types(red_d_full_3m)
red_d_full_6m <- fix_red_types(red_d_full_6m)
red_d_full_12m <- fix_red_types(red_d_full_12m)

# Combine all data frames
red_d_full_012m <- bind_rows(
  red_d_full_3m,
  red_d_full_6m,
  red_d_full_12m
) %>%
  arrange(time_point, system, intervention, study) %>%
  mutate(row_id = row_number())

# CRAVING DATA FRAMES
# Define function to fix the data types for the crav data frames
fix_crav_types <- function(df) {
  df %>%
    mutate(across(
      any_of(c(
        "total_n", "c_n", "e_n",
        "c_mean", "e_mean", "c_sd", "e_sd",
        "SMD", "SE", "LCI", "UCI",
        "lnOR", "SElnOR", "lnLCI", "lnUCI",
        "OR", "OrSE", "OrLCI", "OrUCI",
        "z_value", "p_value"
      )),
      ~ suppressWarnings(as.numeric(.))
    ))
}

# Fix data frames before merging

```

```

crav_d_full_3m <- fix_crav_types(crav_d_full_3m)
crav_d_full_6m <- fix_crav_types(crav_d_full_6m)
crav_d_full_12m <- fix_crav_types(crav_d_full_12m)
# Combine all data frames
crav_d_full_012m <- bind_rows(
  crav_d_full_3m,
  crav_d_full_6m,
  crav_d_full_12m
) %>%
  arrange(time_point, system, intervention, study) %>%
  mutate(row_id = row_number())

```

Secondly, we define a new list reversing the previously defined study labels.

```

# Swap names and values
base_ids <- sub("^([0-9]+)[a-z]_(.*)$", "\\1\\2", names(study_labels))

# Now build reversed mapping: from APA label -> base ID (e.g. "Dubuson et al. (2021)" -> "3_dubuson_2021")
study_labels_reversed <- setNames(base_ids, study_labels)

```

Now we can create a list that determines the correct study ID based on the combination of base ID and time point.

```

# Function to add study ID labels based on time point
time_map <- tribble(
  ~base_id,          ~time_point,      ~study_id,

  # 1 - Chen
  "1_chen_2022",      "6 months",      "1_chen_2022",

  # 2 - Uyl
  "2_uyl_2018",       "0-3 months",    "2a_uyl_2018",
  "2_uyl_2018",       "12 months",     "2b_uyl_2018",

  # 3 - Dubuson
  "3_dubuson_2021",   "0-3 months",    "3c_dubuson_2021",
  "3_dubuson_2021",   "6 months",      "3d_dubuson_2021",
  "3_dubuson_2021",   "12 months",     "3e_dubuson_2021",

  # 4 - Eberl
  "4_eberl_2013",     "12 months",     "4_eberl_2013",

  # 5 - Manning 2022 (+ Garfield 2022)
  "5_manning_2022",   "0-3 months",    "5b_manning_2022",
  "5_manning_2022",   "6 months",      "5c_manning_2022",
  "5_manning_2022",   "12 months",     "5d_manning_2022",

  "5_garfield_2022",   "0-3 months",    "5f_garfield_2022",
  "5_garfield_2022",   "12 months",     "5h_garfield_2022",

  # 6 - Grieder
  "6_grieder_2022",   "0-3 months",    "6_grieder_2022",

  # 7 - Garfield 2026
  "7_garfield_2026",   "0-3 months",    "7c_garfield_2026",

```

```

# 8 - Kvamme
"8_kvamme_2019",      "0-3 months",      "8b_kvamme_2019",

# 9 - Kumar
"9_kumar_2019",      "0-3 months",      "9b_kumar_2019",
"9_kumar_2019",      "6 months",        "9c_kumar_2019",

# 10 - Laurens
"10_laurens_2023",   "0-3 months",      "10b_laurens_2023",
"10_laurens_2023",   "6 months",        "10c_laurens_2023",

# 11 - Manning 2016
"11_manning_2016",   "0-3 months",      "11_manning_2016",

# 12 - Peerenboom
"12_peerenboom_2022", "0-3 months",      "12a_peerenboom_2022",
"12_peerenboom_2022", "12 months",       "12b_peerenboom_2022",

# 13 - Rinck
"13_rinck_2018",     "12 months",       "13a_rinck_2018",

# 14 - Schenkel 2023
"14_schenkel_2023",   "0-3 months",      "14a_schenkel_2023",
"14_schenkel_2023",   "12 months",       "14b_schenkel_2023",

# 15 - Schenkel 2024
"15_schenkel_2024",   "0-3 months",      "15a_schenkel_2024",
"15_schenkel_2024",   "6 months",        "15b_schenkel_2024",
"15_schenkel_2024",   "12 months",       "15c_schenkel_2024",

# 16 - Schoenmakers
"16_schoenmakers_2010", "0-3 months",      "16b_schoenmakers_2010",

# 17 - Stein
"17_stein_2023",      "0-3 months",      "17b_stein_2023",

# 18 - Wiers
"18_wiers_2011",     "0-3 months",      "18a_wiers_2011",
"18_wiers_2011",     "12 months",       "18b_wiers_2011",

# 19 - Spruyt
"19_spruyt_2025",     "0-3 months",      "19a_spruyt_2025",
"19_spruyt_2025",     "6 months",        "19b_spruyt_2025",
"19_spruyt_2025",     "12 months",       "19c_spruyt_2025",

# 20 - Mistarz
"20_mistarz_2025",     "0-3 months",      "20a_mistarz_2025",
"20_mistarz_2025",     "6 months",        "20b_mistarz_2025",

# 21 - Salemink
"21_salemink_2022",   "12 months",       "21_salemink_2022"
)

```

Now we can create a function that applies the correct study labels to each of the three data frames based on the time point and base ID.

```
# Function to apply study labels based on time point
apply_study_labels <- function(df, time_map, study_labels_reversed) {

  df %>%
    # Standardize time_point so it matches the keys in time_map
    mutate(
      time_point = case_when(
        time_point %in% c("0-3 months", "0 - 3 months", "0 -3 months") ~ "0-3 months",
        TRUE ~ time_point
      )
    ) %>%
    # base_id from APA label; if study is already a code, this will be NA
    mutate(base_id = study_labels_reversed[study]) %>%
    # join on base_id + time_point
    left_join(time_map, by = c("base_id", "time_point")) %>%
    mutate(
      # If a mapping exists, use study_id;
      # otherwise keep the original study value (this protects 5d_manning_2022, 5h_garfield_2022, etc.)
      study = if_else(!is.na(study_id), study_id, study)
    ) %>%
    select(-base_id, -study_id)
}
```

Now we can apply this function to all three data frames containing the calculated effect sizes.

```
# Apply the function to each data frame
abs_d_full_012m <- apply_study_labels(abs_d_full_012m, time_map, study_labels_reversed)
crav_d_full_012m <- apply_study_labels(crav_d_full_012m, time_map, study_labels_reversed)
red_d_full_012m <- apply_study_labels(red_d_full_012m, time_map, study_labels_reversed)
```

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 abs 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 main analyses.

```
# Define study IDs for abs
abs_rob <- c(
  "1_chen_2022",
  "2b_uyl_2018",
  "3e_dubuson_2021",
  "4_eberl_2013",
  "5d_manning_2022",
  "6_grieder_2022",
  "8b_kvamme_2019",
  "9c_kumar_2019",
  "11_manning_2016",
  "12b_peerenboom_2022",
  "13a_rinck_2018",
  "14b_schenkel_2023",
  "15c_schenkel_2024",
  "16b_schoenmakers_2010",
  "17b_stein_2023",
)
```

```

      "18b_wiers_2011",
      "19c_spruyt_2025",
      "20b_mistarz_2025",
      "21_salemink_2022"
    )

    # Define study IDs for alcohol reduction
    red_rob <- c(
      "7c_garfield_2026", "10c_laurens_2023"
    )

    # Filter and select only needed columns in the data frame for abs
    abs_d_rob <- abs_d_full_012m %>%
      filter(study %in% abs_rob) %>%
      select(study, SE)

    # Filter and select only needed columns in the data frame for alcohol reduction
    red_d_rob <- red_d_full_012m %>%
      filter(study %in% red_rob) %>%
      select(study, SE)

    # Combine the two sets
    rob_d_se <- bind_rows(abs_d_rob, red_d_rob) %>%
      mutate(SE = suppressWarnings(as.numeric(SE)))

```

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. After this, the relative weights can be added to the final RoB-summary data and exported as a new csv-file named “Weighted RoB Data”.

```

# Remove non-unique reports from rob_d first (so they don't appear in output)
rob_d_unique <- rob_d %>%
  dplyr::filter(!Study %in% c("Garfield et al. (2022)*", "Manning et al. (2021)*", "Peerenboom et al. (2021)*"))

# 1) Build weights keyed by BASE study id from rob_d_se (coded ids like 3c_dubuson_2021)
rob_weights_base <- rob_d_se %>%
  dplyr::mutate(
    base_id = sub("^([0-9]+)[a-z](.*)$", "\\1\\2", study),
    SE = suppressWarnings(as.numeric(SE))
  ) %>%
  dplyr::filter(!is.na(SE), SE > 0) %>%
  dplyr::group_by(base_id) %>%
  dplyr::summarise(weight = sum(1 / (SE^2)), .groups = "drop") %>%
  dplyr::mutate(weight_percent = 100 * weight / sum(weight)) %>%
  dplyr::select(base_id, weight_percent)

# 2) Join weights onto RoB table using APA Study label -> base_id mapping, but do NOT keep helper column
rob_d_weighted <- rob_d_unique %>%
  dplyr::mutate(
    join_base_id = unname(study_labels_reversed[gsub("\\*$", "", Study)])
  ) %>%
  dplyr::left_join(
    rob_weights_base,
    by = c("join_base_id" = "base_id")
  ) %>%

```

```

dplyr::select(-join_base_id)

# 3) Optional diagnostic: show which studies didn't get a weight (if any)
if (any(is.na(rob_d_weighted$weight_percent))) {
  message("These studies did not receive a weight (label mismatch or missing SE):")
  print(rob_d_weighted %>% dplyr::filter(is.na(weight_percent)) %>% dplyr::distinct(Study))
}

# 4) Export
write.csv(rob_d_weighted, here::here("outputs", "data", "Weighted RoB Data.csv"), row.names = FALSE)

```

Now we want to display the percentages, so that these can be reported in the manuscript. We report two versions: the unweighted percentages (each study counts equally) and the weighted percentages (each study weighted by its inverse-variance meta-analytic weight), the latter matching the weighted risk of bias summary figure.

We first calculate the unweighted percentages, where each study contributes equally regardless of its precision.

```

# Unweighted percentages: each study counts equally (simple counts)
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"))
  )

# 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 x 4
## # Groups:   Domain [6]
##   Domain                                Low `Some concerns` High
##   <fct>                                <dbl>         <dbl> <dbl>
## 1 Randomization process                71.4           23.8   4.8
## 2 Deviations from intended interventions  81             19      0
## 3 Missing outcome data                42.9           38.1  19
## 4 Measurement of the outcome           66.7           23.8   9.5
## 5 Selection of the reported result       66.7           33.3   0
## 6 Overall risk of bias                  19             42.9  38.1
```

We then calculate the weighted percentages, where each study is weighted by its inverse-variance meta-analytic weight. These percentages correspond to the weighted risk of bias summary figure and, unlike the unweighted table above, exclude studies that did not contribute a weight.

```
# Weighted percentages: each study weighted by its meta-analytic weight
rob_pct_w <- rob_d_weighted %>%
  dplyr::filter(!is.na(weight_percent)) %>%
  pivot_longer(cols = D1:Overall, names_to = "Domain", values_to = "Judgment") %>%
  mutate(
    Domain = recode(Domain, !!!rob_labels),
    Domain = factor(Domain, levels = domain_order),
    Judgment = factor(Judgment, levels = c("Low", "Some concerns", "High"))
  )

# Sum the study weights per judgement within each domain, then convert to a
# percentage of the domain's total weight
rob_pct_w_summary <- rob_pct_w %>%
  group_by(Domain, Judgment) %>%
  summarise(weight = sum(weight_percent), .groups = "drop") %>%
  group_by(Domain) %>%
  mutate(Percent = round(100 * weight / sum(weight), 1))

# Wide format summary table
rob_pct_w_summary <- rob_pct_w_summary %>%
  select(Domain, Judgment, Percent) %>%
  pivot_wider(names_from = Judgment, values_from = Percent, values_fill = 0)

# Print the ordered weighted summary table
print(rob_pct_w_summary)
```

```
## # A tibble: 6 x 4
## # Groups:   Domain [6]
##   Domain                                Low `Some concerns` High
##   <fct>                                <dbl>         <dbl> <dbl>
## 1 Randomization process                67.6           32    0.4
## 2 Deviations from intended interventions  96.1            3.9   0
## 3 Missing outcome data                45.8           39.6  14.6
## 4 Measurement of the outcome           37.2           53.1   9.7
## 5 Selection of the reported result       42.3           57.7   0
## 6 Overall risk of bias                  16.6           33.3  50.1
```

6.2 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 six manuscript figures (fig_2 through fig_7) plus two supplementary figures. The manuscript figures follow the naming convention “fig_x_”, and as a flow-chart is included as the first figure in the manuscript, the figures start with “fig_2_”. The supplementary figures follow the naming convention “fig_Sx_” and comprise the traffic light plot (S1) and the weighted risk of bias summary (S2).

We first export the traffic light plot of the individual RoB-ratings. As this figure is not used in the manuscript, it is exported as supplementary figure S1.

We then export the unweighted summary of the RoB-assessment across all domains. This is the summary reported in the manuscript, and it will be exported as figure 2.

We also export the weighted summary as supplementary figure S2, where each study is weighted by its inverse-variance meta-analytic weight, so the bars reflect the studies’ contribution to the pooled estimates rather than a simple study count.

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

The first forest plot for our main analysis on short-term effects on abstinence, with the integrated risk of bias ratings, will be exported as figure 3.

```
# Match 3m studies to RoB ratings
rob_3m_match <- rob_d %>%
  dplyr::mutate(Study_clean = gsub("\\*$", "", Study)) %>%
  dplyr::filter(Study_clean %in% abs_metagen_3m$studlab) %>%
  dplyr::distinct(Study_clean, .keep_all = TRUE)
rob_3m_match <- rob_3m_match[match(abs_metagen_3m$studlab, rob_3m_match$Study_clean), ]
# Custom style
abs_metagen_3m_rob <- rob(
  item1 = rob_3m_match$D1,
  item2 = rob_3m_match$D2,
  item3 = rob_3m_match$D3,
  item4 = rob_3m_match$D4,
  item5 = rob_3m_match$D5,
  overall = rob_3m_match$Overall,
  studlab = rob_3m_match$Study_clean,
  tool = "RoB2",
  data = abs_metagen_3m,
  overwrite = TRUE,
  domains = c("D1", "D2", "D3", "D4", "D5"),
  col = c("#02C100", "#E2DF07", "#BF0000"),
  symbols = c("+", "-", "x"),
  legend = FALSE
)
# Domain names
names(abs_metagen_3m_rob$rob) <- gsub("^A$", "D1", names(abs_metagen_3m_rob$rob))
names(abs_metagen_3m_rob$rob) <- gsub("^B$", "D2", names(abs_metagen_3m_rob$rob))
names(abs_metagen_3m_rob$rob) <- gsub("^C$", "D3", names(abs_metagen_3m_rob$rob))
names(abs_metagen_3m_rob$rob) <- gsub("^D$", "D4", names(abs_metagen_3m_rob$rob))
names(abs_metagen_3m_rob$rob) <- gsub("^E$", "D5", names(abs_metagen_3m_rob$rob))
```

```

for (attr_name in c("symbols", "col", "categories")) {
  a <- attr(abs_metagen_3m_rob$rob, attr_name)
  if (!is.null(a) && !is.null(names(a))) {
    names(a) <- gsub("^A$", "D1", names(a))
    names(a) <- gsub("^B$", "D2", names(a))
    names(a) <- gsub("^C$", "D3", names(a))
    names(a) <- gsub("^D$", "D4", names(a))
    names(a) <- gsub("^E$", "D5", names(a))
    attr(abs_metagen_3m_rob$rob, attr_name) <- a
  }
}

# Prepare the final forest plot
tiff(here::here("outputs", "figures", "fig_3_abs_short_forest_rob.tiff"), width = 13.5, height = 7.5, u
forest(abs_metagen_3m_rob,
  prediction = TRUE,
  prediction.subgroup = FALSE,
  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,
  spacing = 1.3,
  fontsize = 12,
  digits = 2,
  xlim = c(1/30, 30),
  at = c(0.03, 0.08, 0.2, 0.5, 1, 2, 5, 12, 30),
  leftcols = c("studlab", "e_abs", "e_n", "c_abs", "c_n"),
  leftlabs = c("Study", "Intv. (events)", "Total (n)",
    "Ctrl. (events)", "Total (n)"),
  rightcols = c("w.random", "effect", "ci"),
  rightlabs = c("Weight (%)", "OR", "95% CI"),
  rob.legend = FALSE,
  colgap.rob = "2mm",
  colgap.rob.overall = "4mm",
  rob.text = ""
)

grid::grid.text("Risk of Bias", x = 0.924, y = 0.952,
  gp = grid::gpar(fontsize = 12, fontface = "bold"))

footer_imp_3m_rob <- paste0(
  "Subgroup effect: z = ", round(abs_zp_subsplit_3m$Implicit$z_value, 2),
  ", p = ", formatC(abs_zp_subsplit_3m$Implicit$p_value, digits = 2),
  ", NNT = ", abs_zp_subsplit_3m$Implicit$NNT
)
footer_exp_3m_rob <- paste0(

```

```

"Subgroup effect: z = ", round(abs_zp_subsplit_3m$Explicit$z_value, 2),
", p = ", formatC(abs_zp_subsplit_3m$Explicit$p_value, digits = 2),
", NNT = ", abs_zp_subsplit_3m$Explicit$NNT
)
footer_over_3m_rob <- paste0(
  "Overall effect: z = ", round(abs_zp_overall_3m$z_value, 2),
  ", p = ", formatC(abs_zp_overall_3m$p_value, digits = 2),
  ", NNT = ", abs_zp_overall_3m$NNT
)

# Adjust the test statistics
grid::grid.text(footer_imp_3m_rob, x = 0.018, y = 0.362, just = "left",
  gp = grid::gpar(fontsize = 11))
grid::grid.text(footer_exp_3m_rob, x = 0.018, y = 0.153, just = "left",
  gp = grid::gpar(fontsize = 11))
grid::grid.text(footer_over_3m_rob, x = 0.018, y = 0.02, just = "left",
  gp = grid::gpar(fontsize = 11))

gl <- grid::grid.ls(print = FALSE)
for (nm in gl$name) {
  if (!grepl("^GRID\\.text", nm)) next
  g <- grid::grid.get(nm)
  if (is.null(g$label)) next
  labs <- as.character(g$label)
  if (length(labs) == 1 && labs %in% c("+", "-", "x")) {
    y_val <- switch(labs, "x" = 0.56, "+" = 0.50, "-" = 0.55)
    x_val <- switch(labs, "x" = 0.52, "+" = 0.51, "-" = 0.49)
    grid::grid.edit(nm, x = unit(x_val, "npc"), y = unit(y_val, "npc"), redraw = FALSE)
  }
}
for (nm in gl$name) {
  if (!grepl("^GRID\\.text", nm)) next
  g <- grid::grid.get(nm)
  if (is.null(g$label)) next
  if (length(as.character(g$label)) == 1 && as.character(g$label) == "0") {
    grid::grid.edit(nm, label = "0A", redraw = FALSE)
  }
}
grid::grid.refresh()
invisible(dev.off())

```

The second forest plot for our main analysis on intermediate-term effects on abstinence, with the integrated risk of bias ratings, will be exported as figure 4.

```

# Re-run metagen with subgroup
abs_metagen_6m <- metagen(TE = lnOR,
  seTE = SElnOR,
  studlab = study,
  data = abs_d_full_6m,
  subgroup = `Cognitive system`,
  sm = "OR",
  method.tau = "REML",
  common = FALSE,
  random = TRUE,

```

```

        method.random.ci = "classic")

# Match 6m studies to RoB ratings
rob_6m_match <- rob_d %>%
  dplyr::mutate(Study_clean = gsub("\\*$", "", Study)) %>%
  dplyr::filter(Study_clean %in% abs_metagen_6m$studlab) %>%
  dplyr::distinct(Study_clean, .keep_all = TRUE)
rob_6m_match <- rob_6m_match[match(abs_metagen_6m$studlab, rob_6m_match$Study_clean), ]

# Custom styling
abs_metagen_6m_rob <- rob(
  item1 = rob_6m_match$D1,
  item2 = rob_6m_match$D2,
  item3 = rob_6m_match$D3,
  item4 = rob_6m_match$D4,
  item5 = rob_6m_match$D5,
  overall = rob_6m_match$Overall,
  studlab = rob_6m_match$Study_clean,
  tool = "RoB2",
  data = abs_metagen_6m,
  overwrite = TRUE,
  domains = c("D1", "D2", "D3", "D4", "D5"),
  col = c("#02C100", "#E2DF07", "#BF0000"),
  symbols = c("+", "-", "x"),
  legend = FALSE
)

# Domain labels
names(abs_metagen_6m_rob$rob) <- gsub("^A$", "D1", names(abs_metagen_6m_rob$rob))
names(abs_metagen_6m_rob$rob) <- gsub("^B$", "D2", names(abs_metagen_6m_rob$rob))
names(abs_metagen_6m_rob$rob) <- gsub("^C$", "D3", names(abs_metagen_6m_rob$rob))
names(abs_metagen_6m_rob$rob) <- gsub("^D$", "D4", names(abs_metagen_6m_rob$rob))
names(abs_metagen_6m_rob$rob) <- gsub("^E$", "D5", names(abs_metagen_6m_rob$rob))
for (attr_name in c("symbols", "col", "categories")) {
  a <- attr(abs_metagen_6m_rob$rob, attr_name)
  if (!is.null(a) && !is.null(names(a))) {
    names(a) <- gsub("^A$", "D1", names(a))
    names(a) <- gsub("^B$", "D2", names(a))
    names(a) <- gsub("^C$", "D3", names(a))
    names(a) <- gsub("^D$", "D4", names(a))
    names(a) <- gsub("^E$", "D5", names(a))
    attr(abs_metagen_6m_rob$rob, attr_name) <- a
  }
}

# Prepare the final forest plot
tiff(here::here("outputs", "figures", "fig_4_abs_inter_forest_rob.tiff"), width = 13, height = 6, units
forest(abs_metagen_6m_rob,
  prediction = TRUE,
  prediction.subgroup = FALSE,
  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,
    spacing         = 1.3,
    fontsize        = 12,
    digits          = 2,
    xlim            = c(1/30, 30),
    at              = c(0.03, 0.08, 0.2, 0.5, 1, 2, 5, 12, 30),
    leftcols        = c("studlab", "e_abs", "e_n", "c_abs", "c_n"),
    leftlabs        = c("Study", "Intv. (events)", "Total (n)",
                        "Ctrl. (events)", "Total (n)"),
    rightcols       = c("w.random", "effect", "ci"),
    rightlabs       = c("Weight (%)", "OR", "95% CI"),
    rob.legend      = FALSE,
    colgap.rob      = "2mm",
    colgap.rob.overall = "4mm",
    rob.text        = ""
  )

  grid::grid.text("Risk of Bias", x = 0.925, y = 0.932,
    gp = grid::gpar(fontsize = 12, fontface = "bold"))

  footer_imp_6m_rob <- paste0(
    "Subgroup effect: z = ", round(abs_zp_subsplit_6m$Implicit$z_value, 2),
    ", p = ", formatC(abs_zp_subsplit_6m$Implicit$p_value, digits = 2),
    ", NNT = ", abs_zp_subsplit_6m$Implicit$NNT
  )
  footer_exp_6m_rob <- paste0(
    "Subgroup effect: z = ", round(abs_zp_subsplit_6m$Explicit$z_value, 2),
    ", p = ", formatC(abs_zp_subsplit_6m$Explicit$p_value, digits = 2),
    ", NNT = ", abs_zp_subsplit_6m$Explicit$NNT
  )
  footer_over_6m_rob <- paste0(
    "Overall effect: z = ", round(abs_zp_overall_6m$z_value, 2),
    ", p = ", formatC(abs_zp_overall_6m$p_value, digits = 2),
    ", NNT = ", abs_zp_overall_6m$NNT
  )
  # Adjust the test statistics
  grid::grid.text(footer_imp_6m_rob, x = 0.015, y = 0.46, just = "left",
    gp = grid::gpar(fontsize = 11))
  grid::grid.text(footer_exp_6m_rob, x = 0.015, y = 0.20, just = "left",
    gp = grid::gpar(fontsize = 11))
  grid::grid.text(footer_over_6m_rob, x = 0.015, y = 0.025, just = "left",
    gp = grid::gpar(fontsize = 11))

  gl <- grid::grid.ls(print = FALSE)
  for (nm in gl$name) {
    if (!grepl("^GRID\\.text", nm)) next

```

```

g <- grid::grid.get(nm)
if (is.null(g$label)) next
labs <- as.character(g$label)
if (length(labs) == 1 && labs %in% c("+", "-", "x")) {
  y_val <- switch(labs, "x" = 0.56, "+" = 0.50, "-" = 0.55)
  x_val <- switch(labs, "x" = 0.52, "+" = 0.51, "-" = 0.49)
  grid::grid.edit(nm, x = unit(x_val, "npc"), y = unit(y_val, "npc"), redraw = FALSE)
}
}
for (nm in gl$name) {
  if (!grepl("^GRID\\.text", nm)) next
  g <- grid::grid.get(nm)
  if (is.null(g$label)) next
  if (length(as.character(g$label)) == 1 && as.character(g$label) == "0") {
    grid::grid.edit(nm, label = "0A", redraw = FALSE)
  }
}
grid::grid.refresh()
invisible(dev.off())

```

The third forest plot for our exploratory analysis on long-term effects on abstinence by intervention type, with the integrated risk of bias ratings, will be exported as figure 5.

```

# Match studies in the 12m-by-intervention meta-analysis to their RoB ratings
rob_12m <- rob_d %>%
  dplyr::mutate(Study_clean = gsub("\\*$", "", Study)) %>%
  dplyr::filter(Study_clean %in% abs_exp_metagen_12m$studlab) %>%
  dplyr::distinct(Study_clean, .keep_all = TRUE)
rob_12m <- rob_12m[match(abs_exp_metagen_12m$studlab, rob_12m$Study_clean), ]

# Attach RoB to the 12m-by-intervention metagen with explicit domain settings
abs_exp_metagen_12m_rob <- rob(
  item1 = rob_12m$D1,
  item2 = rob_12m$D2,
  item3 = rob_12m$D3,
  item4 = rob_12m$D4,
  item5 = rob_12m$D5,
  overall = rob_12m$Overall,
  studlab = rob_12m$Study_clean,
  tool = "RoB2",
  data = abs_exp_metagen_12m,
  overwrite = TRUE,
  domains = c("D1", "D2", "D3", "D4", "D5"),
  col = c("#02C100", "#E2DF07", "#BF0000"),
  symbols = c("+", "-", "x"),
  legend = FALSE
)

# Domain labels
names(abs_exp_metagen_12m_rob$rob) <- gsub("^A$", "D1", names(abs_exp_metagen_12m_rob$rob))
names(abs_exp_metagen_12m_rob$rob) <- gsub("^B$", "D2", names(abs_exp_metagen_12m_rob$rob))
names(abs_exp_metagen_12m_rob$rob) <- gsub("^C$", "D3", names(abs_exp_metagen_12m_rob$rob))
names(abs_exp_metagen_12m_rob$rob) <- gsub("^D$", "D4", names(abs_exp_metagen_12m_rob$rob))
names(abs_exp_metagen_12m_rob$rob) <- gsub("^E$", "D5", names(abs_exp_metagen_12m_rob$rob))
for (attr_name in c("symbols", "col", "categories")) {

```

```

a <- attr(abs_exp_metagen_12m_rob$rob, attr_name)
if (!is.null(a) && !is.null(names(a))) {
  names(a) <- gsub("^A$", "D1", names(a))
  names(a) <- gsub("^B$", "D2", names(a))
  names(a) <- gsub("^C$", "D3", names(a))
  names(a) <- gsub("^D$", "D4", names(a))
  names(a) <- gsub("^E$", "D5", names(a))
  attr(abs_exp_metagen_12m_rob$rob, attr_name) <- a
}
}
# Prepare final forest plot
tiff(here::here("outputs", "figures", "fig_5_abs_long_intv_forest_rob.tiff"), width = 13, height = 8, u
forest(abs_exp_metagen_12m_rob,
  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,
  spacing = 1.3,
  fontsize = 12,
  digits = 2,
  leftcols = c("studlab", "e_abs", "e_n", "c_abs", "c_n"),
  leftlabs = c("Study", "Intv. (events)", "Total (n)",
    "Ctrl. (events)", "Total (n)"),
  rightcols = c("w.random", "effect", "ci"),
  rightlabs = c("Weight (%)", "OR", "95% CI"),
  rob.legend = FALSE,
  colgap.rob = "2mm",
  colgap.rob.overall = "4mm",
  rob.text = ""
)

grid::grid.text("Risk of Bias", x = 0.925, y = 0.955,
  gp = grid::gpar(fontsize = 12, fontface = "bold"))

footer_itbm_12m_rob <- paste0(
  "Subgroup effect: z = ", round(abs_exp_subsplit_12m$ItBM$z_value, 2),
  ", p = ", formatC(abs_exp_subsplit_12m$ItBM$p_value, digits = 2),
  ", NNH = ", abs(abs_exp_subsplit_12m$ItBM$NNT)
)
footer_apbm_12m_rob <- paste0(
  "Subgroup effect: z = ", round(abs_exp_subsplit_12m$ApBM$z_value, 2),
  ", p = ", formatC(abs_exp_subsplit_12m$ApBM$p_value, digits = 2),
  ", NNT = ", abs_exp_subsplit_12m$ApBM$NNT
)

```

```

footer_over_12m_intv_rob <- paste0(
  "Overall effect: z = ", round(abs_exp_zp_overall_12m$z_value, 2),
  ", p = ", formatC(abs_exp_zp_overall_12m$p_value, digits = 2),
  ", NNT = ", abs_exp_zp_overall_12m$NNT
)

# Adjust test statistics
grid::grid.text(footer_itbm_12m_rob, x = 0.0188, y = 0.562, just = "left",
  gp = grid::gpar(fontsize = 11))
grid::grid.text(footer_apbm_12m_rob, x = 0.0188, y = 0.145, just = "left",
  gp = grid::gpar(fontsize = 11))
grid::grid.text(footer_over_12m_intv_rob, x = 0.0188, y = 0.015, just = "left",
  gp = grid::gpar(fontsize = 11))

gl <- grid::grid.ls(print = FALSE)
for (nm in gl$name) {
  if (!grepl("^GRID\\.text", nm)) next
  g <- grid::grid.get(nm)
  if (is.null(g$label)) next
  labs <- as.character(g$label)
  if (length(labs) == 1 && labs %in% c("+", "-", "x")) {
    y_val <- switch(labs, "x" = 0.56, "+" = 0.50, "-" = 0.55)
    x_val <- switch(labs, "x" = 0.52, "+" = 0.51, "-" = 0.49)
    grid::grid.edit(nm, x = unit(x_val, "npc"), y = unit(y_val, "npc"), redraw = FALSE)
  }
}
for (nm in gl$name) {
  if (!grepl("^GRID\\.text", nm)) next
  g <- grid::grid.get(nm)
  if (is.null(g$label)) next
  if (length(as.character(g$label)) == 1 && as.character(g$label) == "0") {
    grid::grid.edit(nm, label = "0A", redraw = FALSE)
  }
}
grid::grid.refresh()
invisible(dev.off())

```

We finally export the combined contour-enhanced funnel plot for abstinence across the three time points (short-, intermediate-, and long-term), assembled with the `cowplot` package. This figure will be exported as figure 6.

```

# Build one contour-enhanced funnel panel (OR scale), no title
make_abs_funnel <- function(meta_obj, egger_obj, y_max = 1.0) {
  df <- data.frame(lnOR = meta_obj$TE, se = meta_obj$seTE)
  pooled <- meta_obj$TE.random
  se_seq <- seq(0, y_max, length.out = 300)
  x_range <- max(abs(df$lnOR), 2.576 * y_max) * 1.15
  x_lo <- -x_range; x_hi <- x_range

  make_poly <- function(z) rbind(
    data.frame(x = z * se_seq, y = se_seq),
    data.frame(x = rev(-z * se_seq), y = rev(se_seq))
  )
}

```

```

egger_label <- paste0("Egger's test: p = ", round(egger_obj$p.value, 3))
or_breaks   <- c(0.25, 0.5, 1, 2, 4)

ggplot() +
  geom_polygon(data = make_poly(2.576), aes(x = x, y = y), fill = "grey55", colour = NA) +
  geom_polygon(data = make_poly(1.960), aes(x = x, y = y), fill = "grey75", colour = NA) +
  geom_polygon(data = make_poly(1.645), aes(x = x, y = y), fill = "white", colour = NA) +
  geom_vline(xintercept = 0, linetype = "dotted", colour = "black", linewidth = 0.5) +
  geom_vline(xintercept = pooled, linetype = "dotted", colour = "black", linewidth = 0.5) +
  geom_point(data = df, aes(x = lnOR, y = se),
             shape = 21, fill = "white", colour = "black", size = 2.5, stroke = 0.7) +
  annotation_custom(
    grid::textGrob(
      label = egger_label,
      x = unit(0.995, "npc"), y = unit(0.985, "npc"),
      hjust = 1, vjust = 1,
      gp = grid::gpar(fontsize = 15, col = "black")
    )
  ) +
  scale_x_continuous(
    breaks = log(or_breaks), labels = or_breaks,
    limits = c(x_lo, x_hi), name = "Odds Ratio"
  ) +
  scale_y_reverse(
    name = "Standard Error",
    limits = c(y_max, 0),
    breaks = seq(0, y_max, by = 0.2),
    expand = expansion(mult = c(0.02, 0))
  ) +
  theme_classic(base_size = 16) +
  theme(
    panel.background = element_rect(fill = "white"),
    plot.background  = element_rect(fill = "white", colour = "black", linewidth = 0.5),
    axis.line        = element_line(colour = "black", linewidth = 0.4),
    axis.ticks       = element_line(colour = "black", linewidth = 0.4),
    axis.title       = element_text(size = 16),
    axis.text        = element_text(size = 15),
    plot.margin      = margin(18, 14, 10, 10)
  )
}

# Wrap a panel in a ggdraw canvas: label occupies top 12%, funnel the rest
label_panel <- function(p, label_text, label_x = 0.1, label_hjust = 0) {
  ggdraw() +
    draw_plot(p, x = 0.02, y = 0, width = 0.96, height = 0.86) +
    draw_label(label_text, x = label_x, y = 0.97,
              hjust = label_hjust, vjust = 1,
              fontface = "bold", size = 20)
}

p_abs_3m <- make_abs_funnel(abs_pupbias_3m, abs_egger_3m)
p_abs_6m <- make_abs_funnel(abs_metagen_6m, abs_egger_6m)
p_abs_12m <- make_abs_funnel(abs_metagen_12m, abs_egger_12m)

```

```

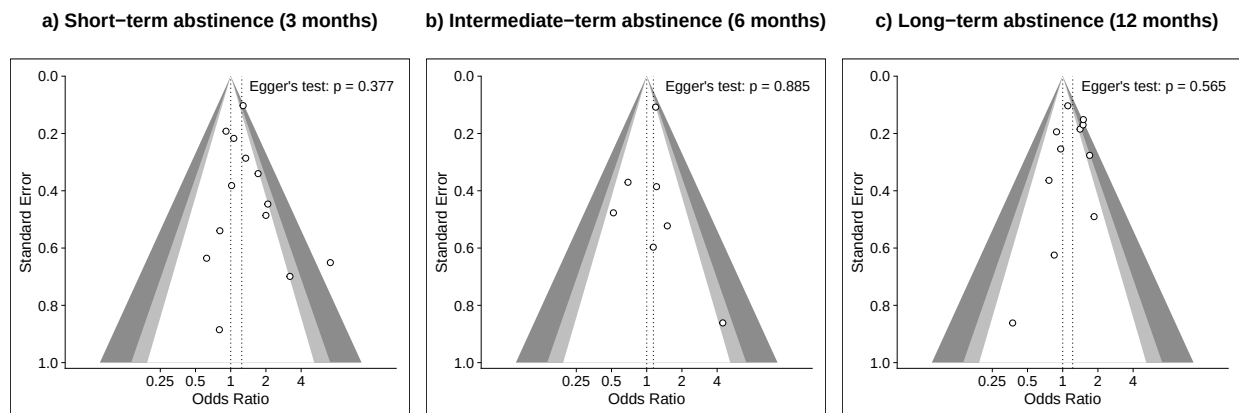
lp_3m <- label_panel(p_abs_3m, "a) Short-term abstinence (3 months)",
                     label_x = 0.5, label_hjust = 0.5)
lp_6m <- label_panel(p_abs_6m, "b) Intermediate-term abstinence (6 months)",
                     label_x = 0.5, label_hjust = 0.5)
lp_12m <- label_panel(p_abs_12m, "c) Long-term abstinence (12 months)")

combined_abs_funnel <- plot_grid(lp_3m, lp_6m, lp_12m, nrow = 1, rel_widths = c(1, 1, 1))

# Export as high-resolution tiff (Figure 6 in final manuscript)
tiff(here::here("outputs", "figures", "fig_6_abs_funnel_combined.tiff"), width = 18, height = 6,
     units = "in", res = 600)
print(combined_abs_funnel)
invisible(dev.off())

# Display inline
combined_abs_funnel

```



We finally export the remaining forest plots for the supplementary material. These are exported as simple forest plots (without the integrated risk of bias ratings), in the order they are produced in the report, and are numbered as supplementary figures S3 through S10.

```

# Sensitivity-analysis forest plot, short-term abstinence (Figure S3)
tiff(here::here("outputs", "figures", "fig_S3_abs_short_sens_forest.tiff"), width = 12, height = 7, uni
forest(abs_sens_metagen_3m,
      prediction = TRUE,
      prediction.subgroup = FALSE,
      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,
        xlim = c(1/30, 30),
        at = c(0.03, 0.08, 0.2, 0.5, 1, 2, 5, 12, 30),
        leftcols = c("studlab", "e_abs", "e_n", "c_abs", "c_n"),
        leftlabs = c("Study", "Intv. (events)", "Total (n)", "Ctrl. (events)", "Total (n)"),
        rightcols = c("w.random", "effect", "ci"),
        rightlabs = c("Weight (%)", "OR", "95% CI")
    )
invisible(dev.off())

# Forest plot, short-term alcohol reduction (Figure S4)
tiff(here::here("outputs", "figures", "fig_S4_red_short_forest.tiff"), width = 11, height = 3, units = "cm")
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"
    )
invisible(dev.off())

# Forest plot, short-term craving (Figure S5)
tiff(here::here("outputs", "figures", "fig_S5_crav_short_forest.tiff"), width = 11, height = 5, units = "cm")
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"
  )
invisible(dev.off())

```

Forest plot, intermediate-term alcohol reduction (Figure S6)

```

tiff(here::here("outputs", "figures", "fig_S6_red_inter_forest.tiff"), width = 12, height = 3.5, units = "cm")
forest(red_metagen_6m,
  prediction = TRUE,
  layout = "BMJ",
  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_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"
)
invisible(dev.off())

```

Forest plot, intermediate-term craving (Figure S7)

```

tiff(here::here("outputs", "figures", "fig_S7_crav_inter_forest.tiff"), width = 11, height = 3, units = "cm")
forest(crav_metagen_6m,
  prediction = TRUE,
  layout = "BMJ",
  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"
  )
invisible(dev.off())

```

Forest plot, short-term abstinence by intervention type (Figure S8)

```

tiff(here::here("outputs", "figures", "fig_S8_abs_short_intv_forest.tiff"), width = 12, height = 10, un
forest(abs_exp_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",
  rightcols = c("w.random", "effect", "ci"),
  rightlabs = c("Weight (%)", "OR", "95% CI")
)
invisible(dev.off())

```

Forest plot, short-term craving by intervention type (Figure S9)

```

tiff(here::here("outputs", "figures", "fig_S9_crav_short_intv_forest.tiff"), width = 12, height = 8, un
forest(crav_exp_metagen_3m,
  prediction = TRUE,
  prediction.subgroup = TRUE,
  layout = "BMJ",
  overall = TRUE,
  subgroup = 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"
)
invisible(dev.off())

```

Forest plot, intermediate-term abstinence by intervention type (Figure S10)

```

tiff(here::here("outputs", "figures", "fig_S10_abs_inter_intv_forest.tiff"), width = 12, height = 7, un
forest(abs_exp_metagen_6m,
      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",
      rightcols = c("w.random", "effect", "ci"),
      rightlabs = c("Weight (%)", "OR", "95% CI")
)
invisible(dev.off())

```

7. 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.

```
# Inline summary table of the packages used. This also writes the
# grateful-refs.bib bibliography (into assets/) that the report cites from.
cite_packages(output = "table", out.dir = here::here("assets"))
```

##	Package	Version	Citation
## 1	base	4.6.1	@base
## 2	devtools	2.5.2	@devtools
## 3	gtable	0.3.6	@gtable
## 4	here	1.0.2	@here
## 5	knitr	1.51	@knitr2014; @knitr2015; @knitr2025
## 6	renv	1.2.3	@renv
## 7	rmarkdown	2.31	@rmarkdown2018; @rmarkdown2020; @rmarkdown2026
## 8	robvis	0.3.0.900	@robvis
## 9	tidyverse	2.0.0	@tidyverse

```
# Standalone grateful report (package citations) saved to reports/
cite_packages(output = "file", out.format = "docx", out.dir = here::here("reports"))
```

```
## [1] "/Users/NM/Documents/Work/Syddansk Universitet/Neuropsychological Interventions/Data Analyses/Me
```

8. 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%3C3127::AID-SIM784%3E3.0.CO;2-M](https://doi.org/10.1002/1097-0258(20001130)19:22%3C3127::AID-SIM784%3E3.0.CO;2-M).
- Grieder, Matthias, Leila M. Soravia, Raphaela M. Tschuemperlin, et al. 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>.
- 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, et al. 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, 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>.