Linear regression - interaction between treatment and IWI - fine motor score

Pearl Ante-Testard

2024-12-16

```
source(here::here("R", "0-config.R"))
```

Fit linear regression with the interaction term between arms_maternal and IWI.

Here, we used lm_robust() to be able to account for clustering in the data when applicable.

```
inter <- df_analysis_fine %>%
  split(.$study) %>%
  map_dfr(function(df_analysis_fine) {
    study_name <- unique(df_analysis_fine$study)
    results <- list()

    if (study_name %in% c("DOSE", "GHANA", "HAITI", "DYADG", "DYADM")) {
      model <- lm_robust(d10_finemotor_el ~ iwi + arms_maternal + iwi * arms_maternal,
                         data = df_analysis_fine)

      beta_interaction_treat_iwi <- coef(model)["iwi:arms_maternalSQ-LNS"]
      model_summary <- summary(model)
      ci_lower <- model_summary$coefficients["iwi:arms_maternalSQ-LNS", "CI Lower"]
      ci_upper <- model_summary$coefficients["iwi:arms_maternalSQ-LNS", "CI Upper"]
      se <- model_summary$coefficients["iwi:arms_maternalSQ-LNS", "Std. Error"]

      results$intervention <- list(inter_treat_iwi = beta_interaction_treat_iwi,
                                   ci_low = ci_lower, ci_up = ci_upper, se = se)

    } else {
      model <- lm_robust(d10_finemotor_el ~ iwi + arms_maternal + iwi * arms_maternal,
                         data = df_analysis_fine, clusters = cluster)

      beta_interaction_treat_iwi <- coef(model)["iwi:arms_maternalSQ-LNS"]
      model_summary <- summary(model)
      ci_lower <- model_summary$coefficients["iwi:arms_maternalSQ-LNS", "CI Lower"]
      ci_upper <- model_summary$coefficients["iwi:arms_maternalSQ-LNS", "CI Upper"]
      se <- model_summary$coefficients["iwi:arms_maternalSQ-LNS", "Std. Error"]
```

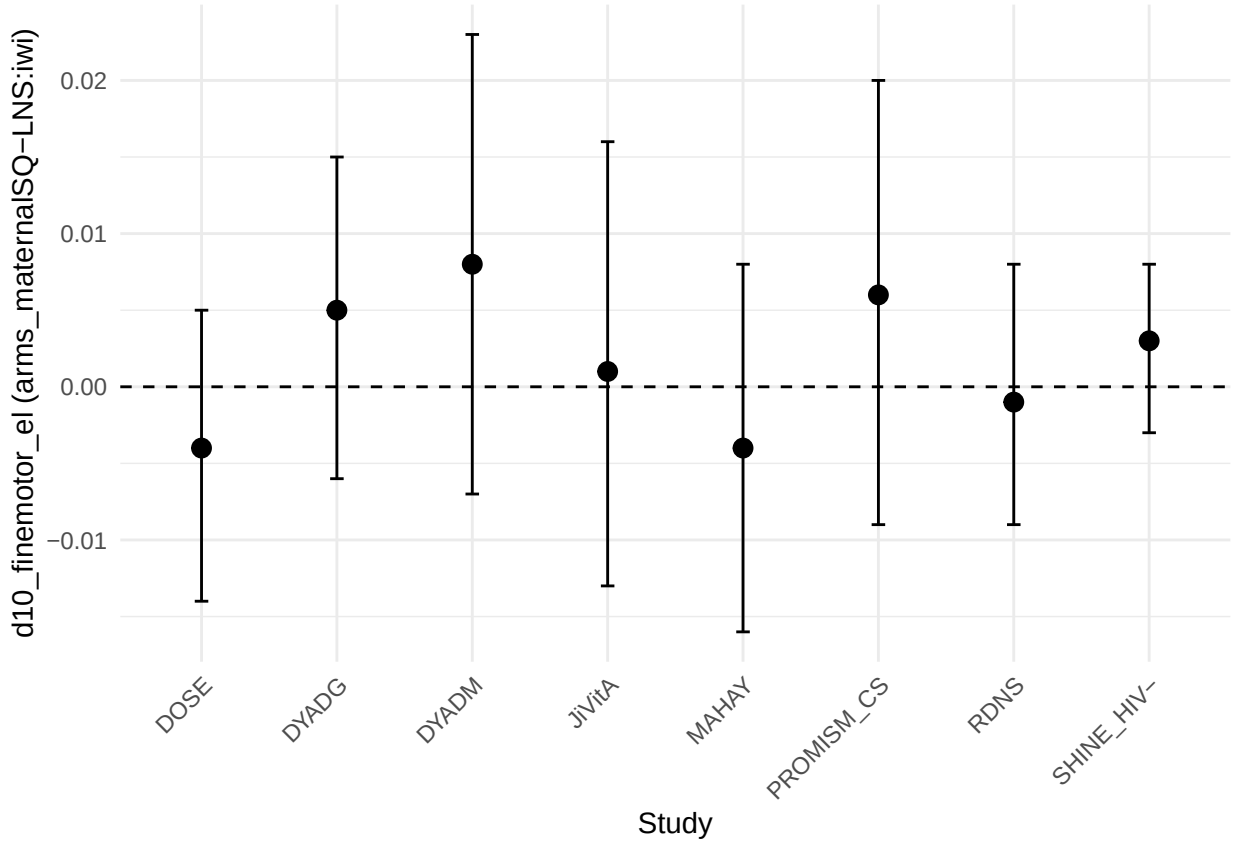
```
    results$intervention <- list(inter_treat_iwi = beta_interaction_treat_iwi,
                                 ci_low = ci_lower, ci_up = ci_upper, se = se)
  }

  # Create a data frame with the results
  inter_tab <- data.frame(
    study = study_name,
    inter_treat_iwi = round(results$intervention$inter_treat_iwi, 3),
    ci_low = round(results$intervention$ci_low, 3),
    ci_up = round(results$intervention$ci_up, 3),
    se = round(results$intervention$se, 3)
  )

  return(inter_tab)
})

print(inter)
```

```
##                                study inter_treat_iwi ci_low ci_up    se
## iwi:arms_maternalSQ-LNS...1       DOSE          -0.004 -0.014 0.005 0.005
## iwi:arms_maternalSQ-LNS...2      DYADG           0.005 -0.006 0.015 0.005
## iwi:arms_maternalSQ-LNS...3      DYADM           0.008 -0.007 0.023 0.008
## iwi:arms_maternalSQ-LNS...4     JiVitA           0.001 -0.013 0.016 0.007
## iwi:arms_maternalSQ-LNS...5      MAHAY          -0.004 -0.016 0.008 0.006
## iwi:arms_maternalSQ-LNS...6 PROMISM_CS           0.006 -0.009 0.020 0.007
## iwi:arms_maternalSQ-LNS...7       RDNS          -0.001 -0.009 0.008 0.004
## iwi:arms_maternalSQ-LNS...8 SHINE_HIV-           0.003 -0.003 0.008 0.003
```

Plotting the study slopes (arms_maternalSQ-LNS:iwi) and their confidence intervals

Here, we plotted the study slopes and their confidence intervals for the interaction term between arms_maternal and IWI for visualization.

```
# Plot the study slope and its confidence interval
ggplot(inter, aes(x = study, y = inter_treat_iwi)) +
  geom_point(size = 3) +
  geom_errorbar(aes(ymin = ci_low, ymax = ci_up), width = 0.1) +
  labs(x = "Study", y = "d10_finemotor_el (arms_maternalSQ-LNS:iwi)") +
  geom_hline(yintercept = 0, linetype = "dashed") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

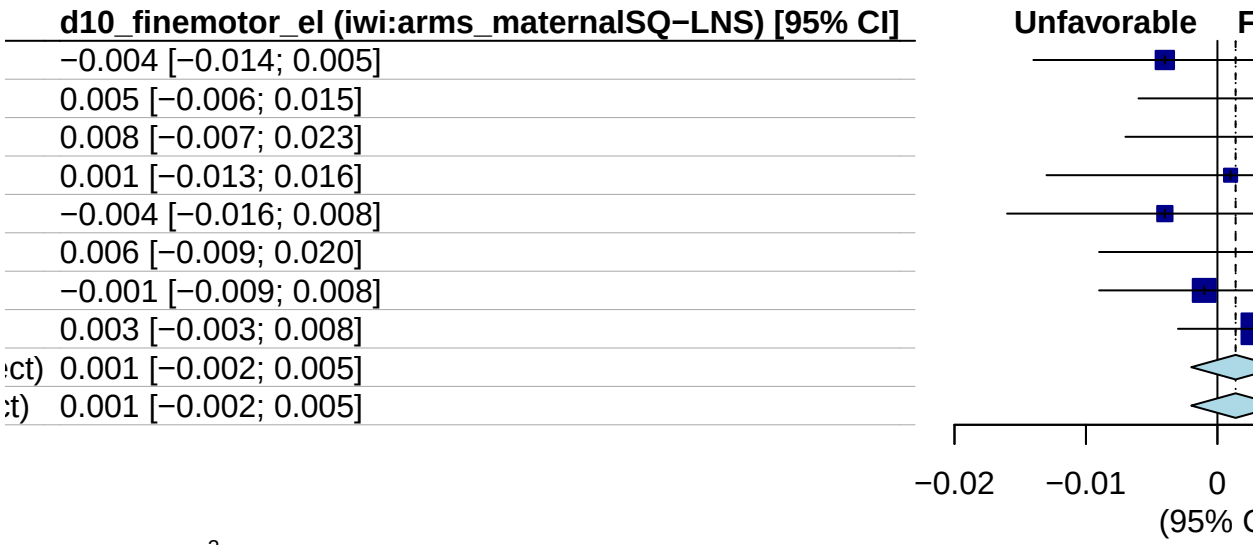


Meta-analysis of the slopes (arms_maternalSQ-LNS:iwi) and their standard errors/confidence intervals from the original linear regression models

Using the slope/coefficients from the interaction term and standard error from the model (iwi:arms_maternalSQ-LNS), we performed a random-effect meta-analysis to get the pooled slope and standard error.

```
# Define the tau.method
tau.method <- "REML"

meta1 <- metagen(
    TE = inter$inter_treat_iwi,
    seTE = inter$se,
    lower = inter$ci_low,
    upper = inter$ci_up,
    method.tau = tau.method,
    hakn = FALSE,
    studlab = inter$study
  )

forest_meta <- forest(meta1,
```

```
        layout = "JAMA",
        leftlabs = c("Study", "d10_finemotor_el (iwi:arms_maternalSQ-LNS) [95% CI]"),
        label.right = "Favorable",
        label.left = "Unfavorable",
        digits = 3,
        prediction.subgroup = inter$arms_maternal)
```



4.25 (P = .75), I^2 = 0%

```
forest_meta
```

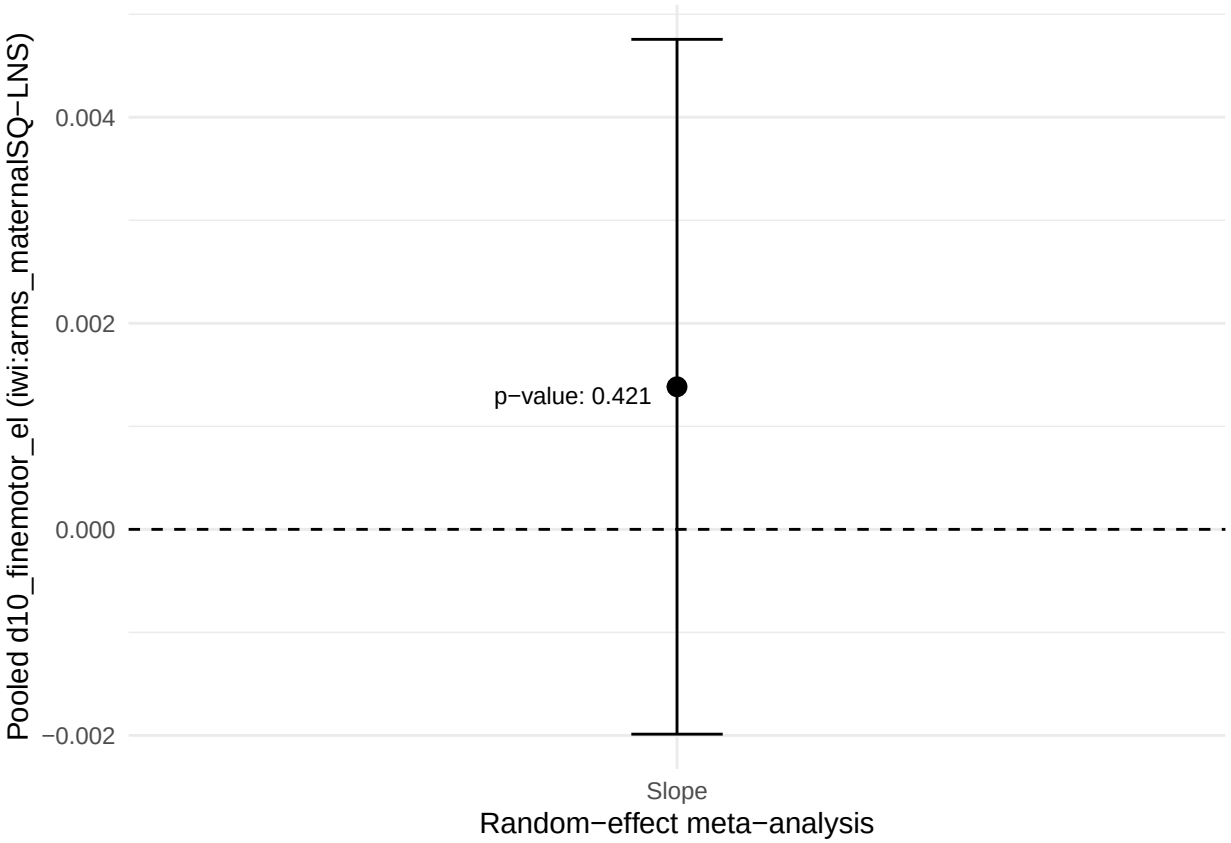
```
## $xlim
## [1] -0.023  0.023
##
## $addrows.below.overall
## [1] 3
##
## $colgap
## [1] 2mm
##
## $colgap.left
## [1] 2mm
##
## $colgap.right
## [1] 2mm
##
## $colgap.studlab
```

```
## [1] 2mm
##
## $colgap.forest
## [1] 2mm
##
## $colgap.forest.left
## [1] 2mm
##
## $colgap.forest.right
## [1] 2mm
##
## $studlab
## [1] "DOSE"       "DYADG"      "DYADM"      "JiVitA"     "MAHAY"      "PROMISM_CS"
## [7] "RDNS"       "SHINE_HIV-"
##
## $TE.format
##  [1] ""        ""        ""        "-0.0040" "0.0050"  "0.0080"  "0.0010"  "-0.0040"
##  [9] "0.0060"  "-0.0010" "0.0030"
##
## $seTE.format
##  [1] ""       ""       ""       "0.0050" "0.0050" "0.0080" "0.0070" "0.0060" "0.0070"
## [10] "0.0040" "0.0030"
##
## $cluster.format
## [1] "" "" ""
##
## $effect.format
##  [1] "0.001"  "0.001"  ""       "-0.004" "0.005"  "0.008"  "0.001"  "-0.004" "0.006"
## [10] "-0.001" "0.003"
##
## $ci.format
##  [1] "[-0.002; 0.005]" "[-0.002; 0.005]" ""                "[-0.014; 0.005]"
##  [5] "[-0.006; 0.015]" "[-0.007; 0.023]" "[-0.013; 0.016]" "[-0.016; 0.008]"
##  [9] "[-0.009; 0.020]" "[-0.009; 0.008]" "[-0.003; 0.008]"
##
## $effect.ci.format
##  [1] "0.001 [-0.002; 0.005]"  "0.001 [-0.002; 0.005]"  ""
##  [4] "-0.004 [-0.014; 0.005]" "0.005 [-0.006; 0.015]"  "0.008 [-0.007; 0.023]"
##  [7] "0.001 [-0.013; 0.016]"  "-0.004 [-0.016; 0.008]" "0.006 [-0.009; 0.020]"
## [10] "-0.001 [-0.009; 0.008]" "0.003 [-0.003; 0.008]"
##
## [[17]]
## NULL
##
## [[18]]
## NULL
##
## $figheight
##   total_height total_rows height_per_row spacing
## 1          3.6         18            0.2       1
##
## $leftcols
## [1] "col.studlab"   "col.effect.ci"
##
```

```
## $leftlabs
## [1] "Study"
## [2] "d10_finemotor_el (iwi:arms_maternalSQ-LNS) [95% CI]"
##
## $rightcols
## [1] "col."
##
## $rightlabs
## NULL
```

Plotting the pooled slope (iwi:arms_maternalSQ-LNS) and its confidence interval

```
laz_eff <- c(meta1$TE.random, meta1$upper.random, meta1$lower.random,
             meta1$seTE.random, meta1$Q, meta1$df.Q, meta1$I2, meta1$tau2,
            meta1$pval.random)
laz_eff_df <- as.data.frame(laz_eff)
laz_eff_df$meta <- c("RE.meta", "RE.meta.upper", "RE.meta.lower", "RE.se",
                     "RE.Q", "RE.df", "RE.I2", "RE.tau2", "pval.random")

laz_eff_df_wide <- laz_eff_df %>%
                  pivot_wider(names_from = meta, values_from = laz_eff)
laz_eff_df_wide$term <- "Slope"
#laz_eff_df_wide$study <- "Pooled"

# Plot the pooled slope and its confidence interval
pooled_slope_plot <- ggplot(aes(x = term, y = RE.meta), data = laz_eff_df_wide) +
  geom_point(size = 3) +
  geom_errorbar(aes(ymin = RE.meta.lower, ymax = RE.meta.upper), width = 0.1) +
  labs(x = "Random-effect meta-analysis", y = "Pooled d10_finemotor_el (iwi:arms_maternalSQ-LNS)") +
  geom_hline(yintercept = 0, linetype = "dashed") +
  theme_minimal()  +
    annotate("text", x = 0.8, y = max(laz_eff_df_wide$RE.meta),
             label = paste("p-value:",
                           round(unique(laz_eff_df_wide$pval.random), 3)),
             hjust = 0, vjust = 1, size = 3, color = "black")
pooled_slope_plot
```



Predictions using the model fit from the linear regression

Using the model above, we generated predictions for a range of IWI values (0-70) for both study arms with confidence intervals. To do this, we first split the data by study, fit the model, and then generated predictions for each study separately. We also extracted the p-value for the interaction term (iwi:arms_maternalSQ-LNS) from the model summary and added it to the plot as an annotation. To make predictions for the control group, we set the arms_maternal variable to "Control" in the new data frame. To make predictions for the intervention group, we set the arms_maternal variable to "SQ-LNS" in the new data frame.

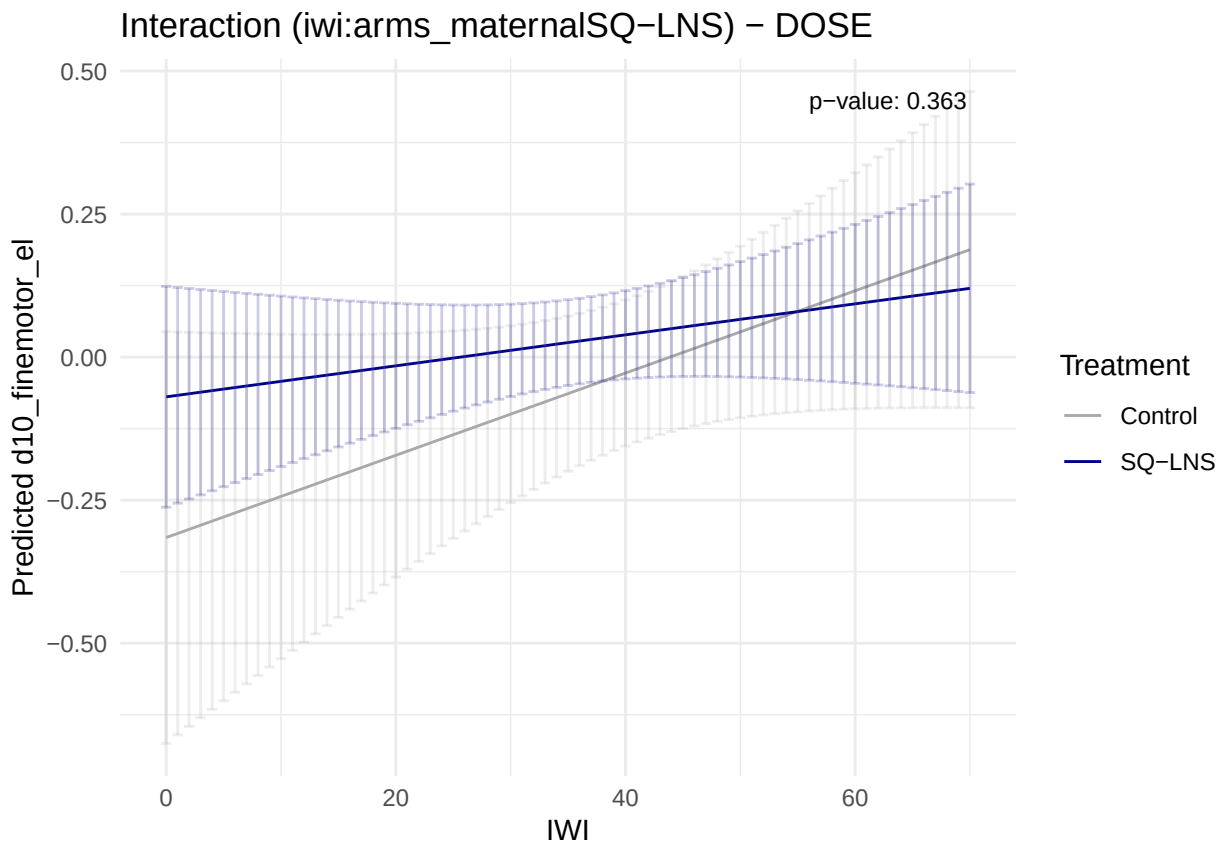
The interaction p-values are from the linear regression models with the interaction term (iwi:arms_maternalSQ-LNS). The p-values indicate whether the effect of IWI on wlz_el differs significantly between the SQ-LNS and control groups.

```
# Define a range of iwi values for prediction
iwi_values <- seq(min(df_analysis_fine$iwi), by = 1, length.out = 71)

# Fit the models and generate predictions for both groups with confidence intervals
predictions <- df_analysis_fine %>%
  split(.$study) %>%
```

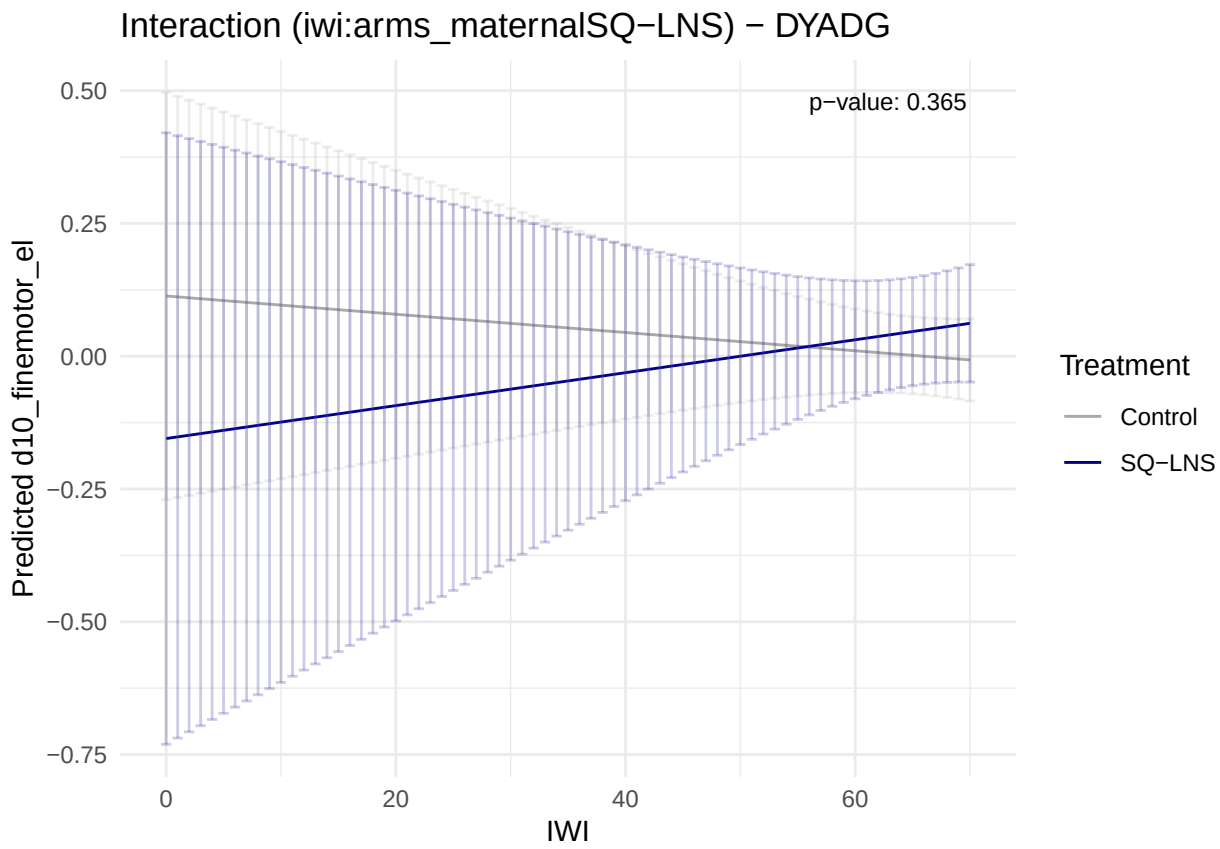
```
  map_dfr(function(df_analysis_fine) {
    study_name <- unique(df_analysis_fine$study)

    if (study_name %in% c("DOSE", "GHANA", "HAITI", "DYADG", "DYADM")) {
      model <- lm_robust(d10_finemotor_el ~ iwi + arms_maternal + iwi * arms_maternal,
                         data = df_analysis_fine)
    } else {
      model <- lm_robust(d10_finemotor_el ~ iwi + arms_maternal + iwi * arms_maternal,
                         data = df_analysis_fine, clusters = cluster)
    }

    # Extract the p-value for the interaction term
    p_value <- summary(model)$coefficients["iwi:arms_maternalSQ-LNS", "Pr(>|t|)"]

    # Generate predictions for a range of iwi values for both groups with confidence intervals
    new_data_SQLNS <- expand.grid(iwi = iwi_values, arms_maternal = "SQ-LNS")
    pred_SQLNS <- predict(model, newdata = new_data_SQLNS, interval = "confidence", level = 0.95)
    new_data_SQLNS <- cbind(new_data_SQLNS, pred_SQLNS)
    new_data_SQLNS$study <- study_name
    new_data_SQLNS$p_value <- p_value
    new_data_SQLNS$group <- "SQ-LNS"

    new_data_control <- expand.grid(iwi = iwi_values, arms_maternal = "Control")
    pred_control <- predict(model, newdata = new_data_control, interval = "confidence", level = 0.95)
    new_data_control <- cbind(new_data_control, pred_control)
    new_data_control$study <- study_name
    new_data_control$p_value <- p_value
    new_data_control$group <- "Control"

    return(bind_rows(new_data_control, new_data_SQLNS))
  })

# Plot the interaction term for each study separately
plots <- predictions %>%
  split(.$study) %>%
  map(function(data) {
    ggplot(data, aes(x = iwi, y = fit.fit, color = arms_maternal)) +
      geom_line() +
      geom_errorbar(aes(ymin = fit.lwr, ymax = fit.upr, fill = group), alpha = 0.2) +
      scale_color_manual(values = c('darkgrey','darkblue')) +
  scale_fill_manual(values = c('darkgrey','darkblue')) +
      theme_minimal() +
      labs(title = paste("Interaction (iwi:arms_maternalSQ-LNS) -", unique(data$study)),
           x = "IWI",
           y = "Predicted d10_finemotor_el",
           color = "Treatment",
           fill = "Treatment") +
      annotate("text", x = max(data$iwi) * 0.8, y = max(data$fit.upr),
               label = paste("p-value:", round(unique(data$p_value), 3)),
               hjust = 0, vjust = 1, size = 3, color = "black")
  })

# Print the plots
```

```
plots
```

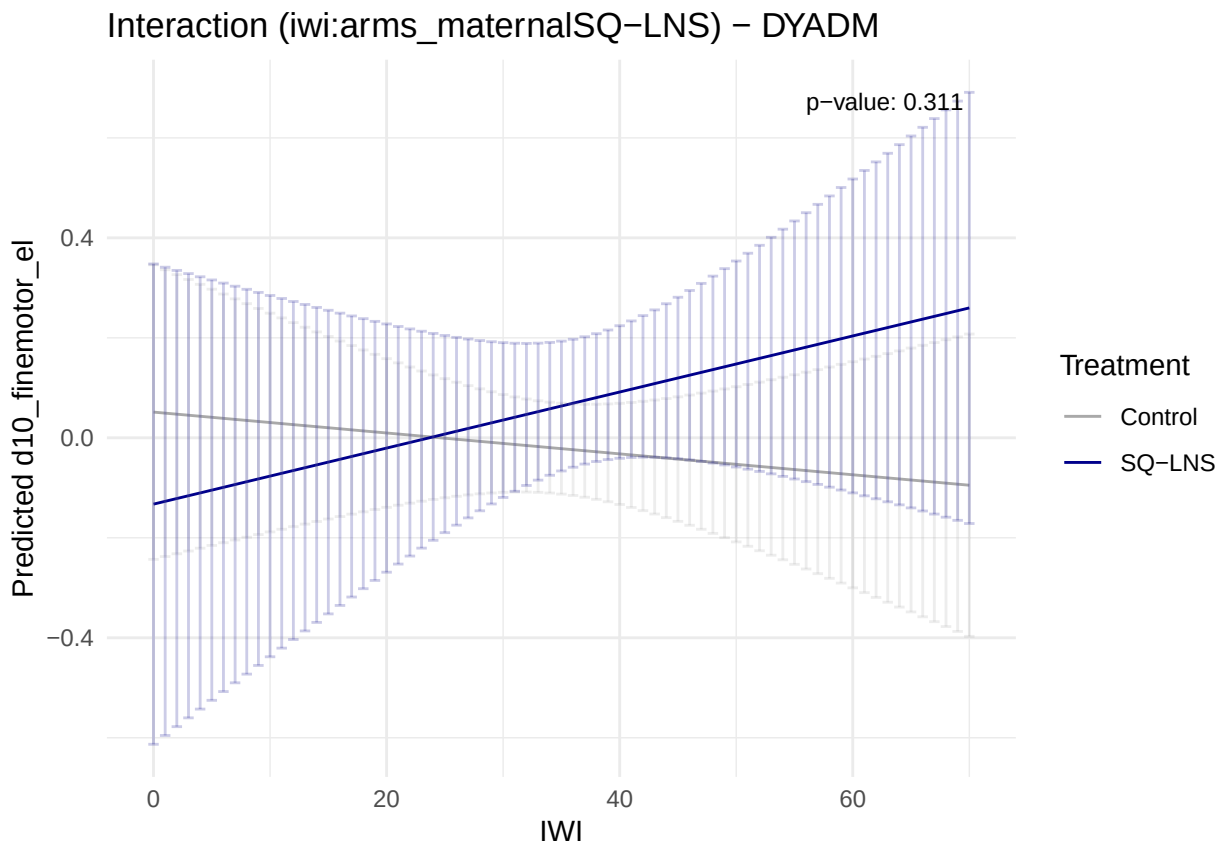
```
## $DOSE
```



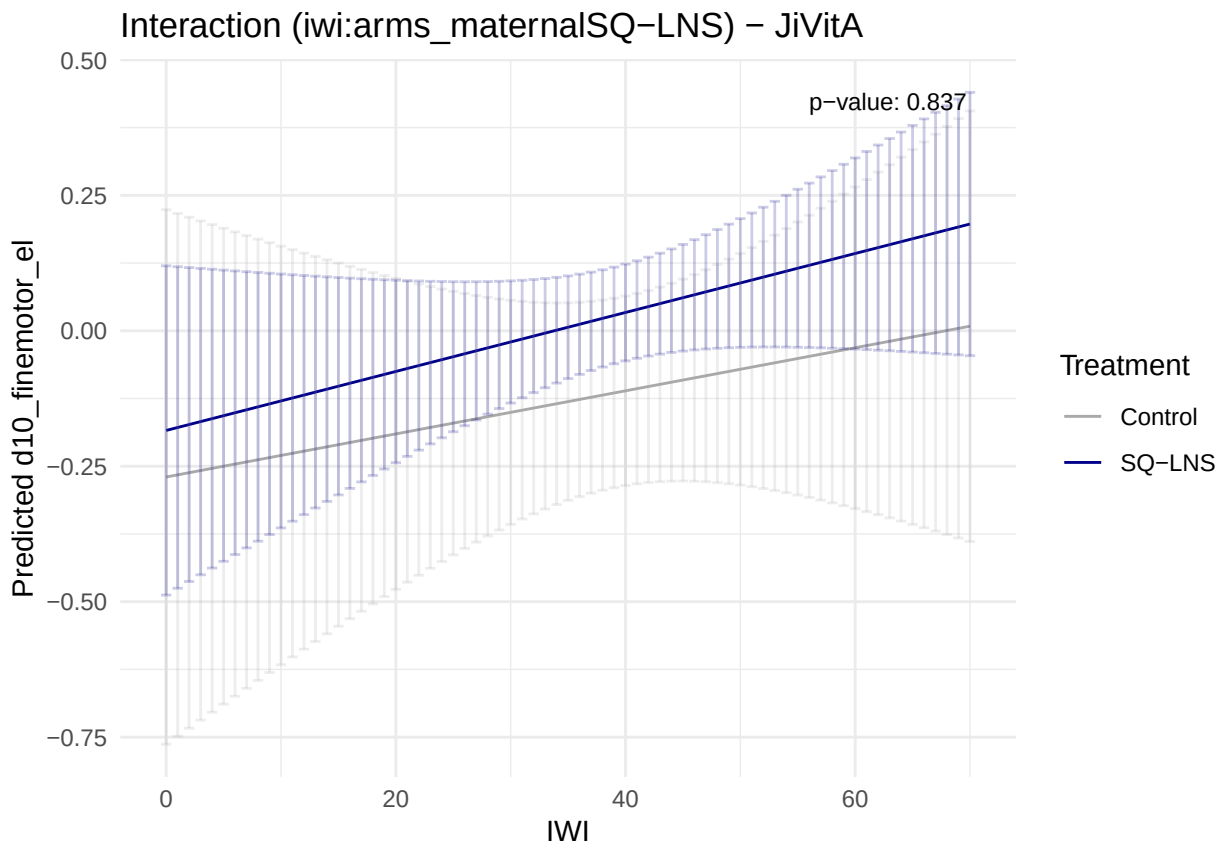

```
##
## $DYADG
```


Interaction (iwi:arms_maternalSQ−LNS) − DYADG
p−value: 0.365
Predicted d10_finemotor_el
0.50
0.25
0.00
−0.25
−0.50
−0.75
0
20
40
60
IWI
Treatment
Control
SQ−LNS

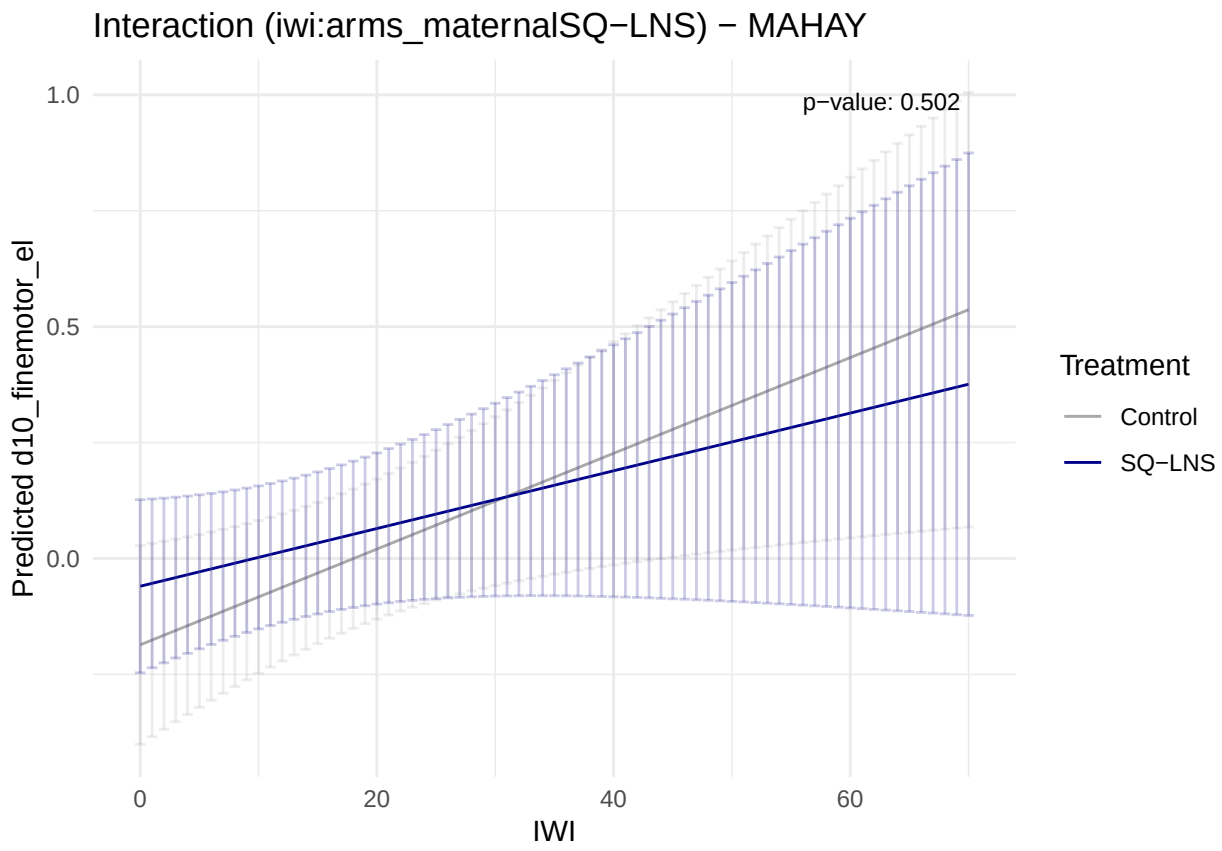
```
##
## $DYADM
```


Interaction (iwi:arms_maternalSQ−LNS) − DYADM
p−value: 0.311
Predicted d10_finemotor_el
0.4
0.0
−0.4
0
20
40
60
IWI
Treatment
Control
SQ−LNS

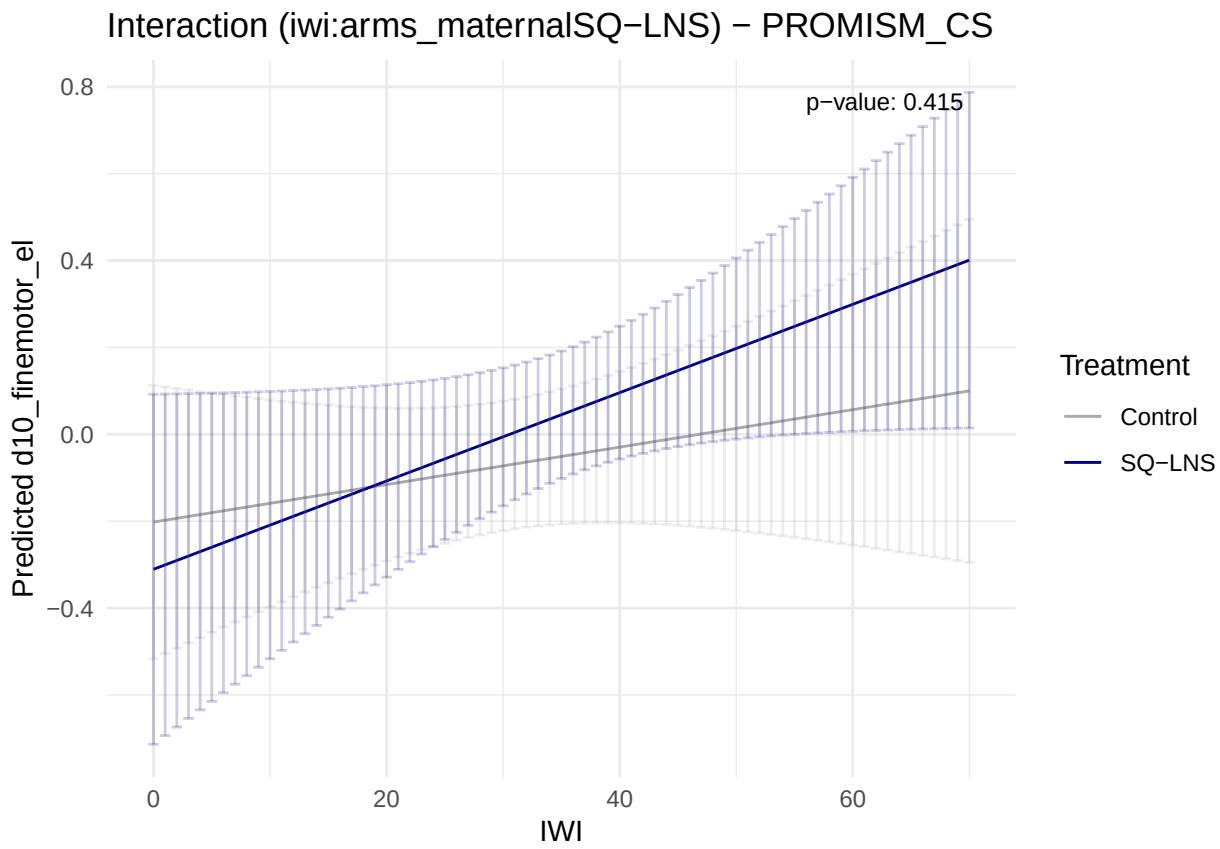
```
##
## $JiVitA
```


Interaction (iwi:arms_maternalSQ−LNS) − JiVitA
p−value: 0.837
Predicted d10_finemotor_el
0.50
0.25
0.00
−0.25
−0.50
−0.75
0
20
40
60
IWI
Treatment
Control
SQ−LNS

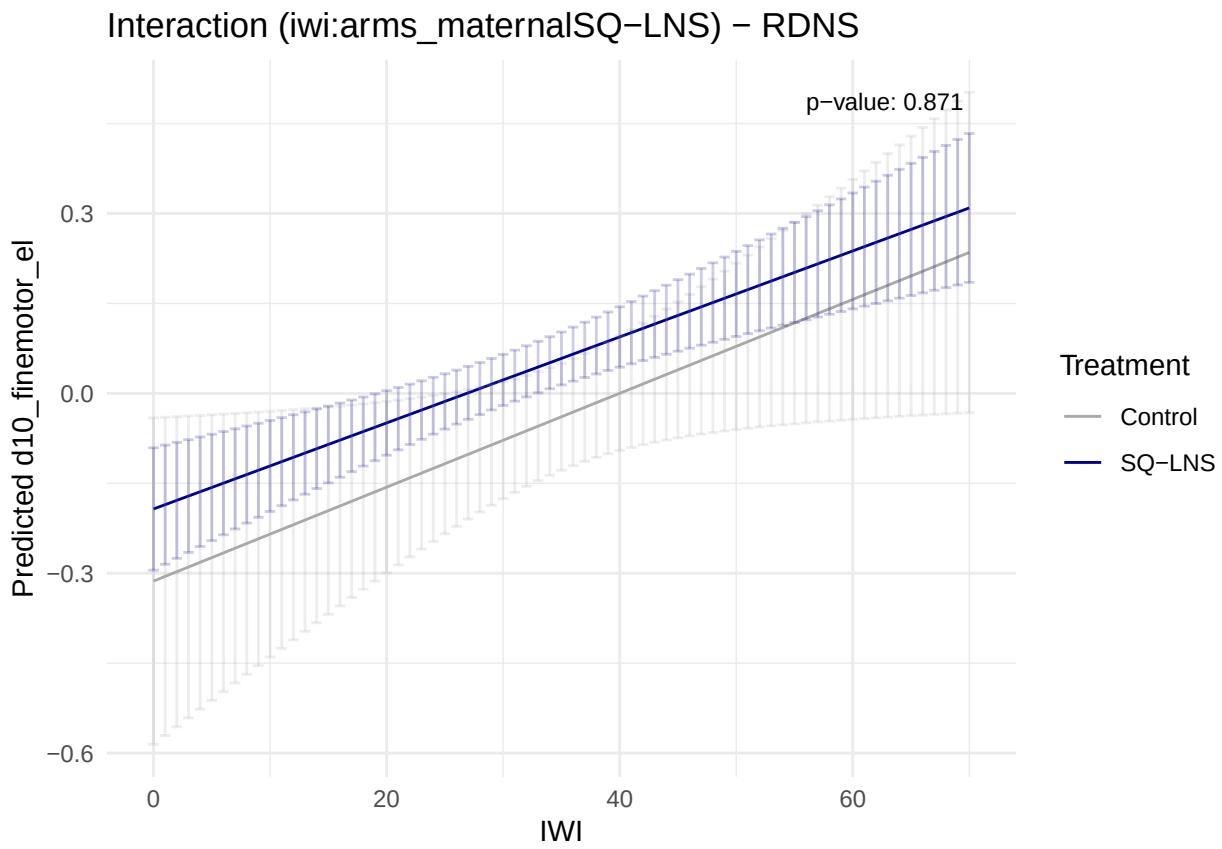
```
##
## $MAHAY
```


Interaction (iwi:arms_maternalSQ−LNS) − MAHAY
p−value: 0.502
Predicted d10_finemotor_el
1.0
0.5
0.0
0
20
40
60
IWI
Treatment
Control
SQ−LNS

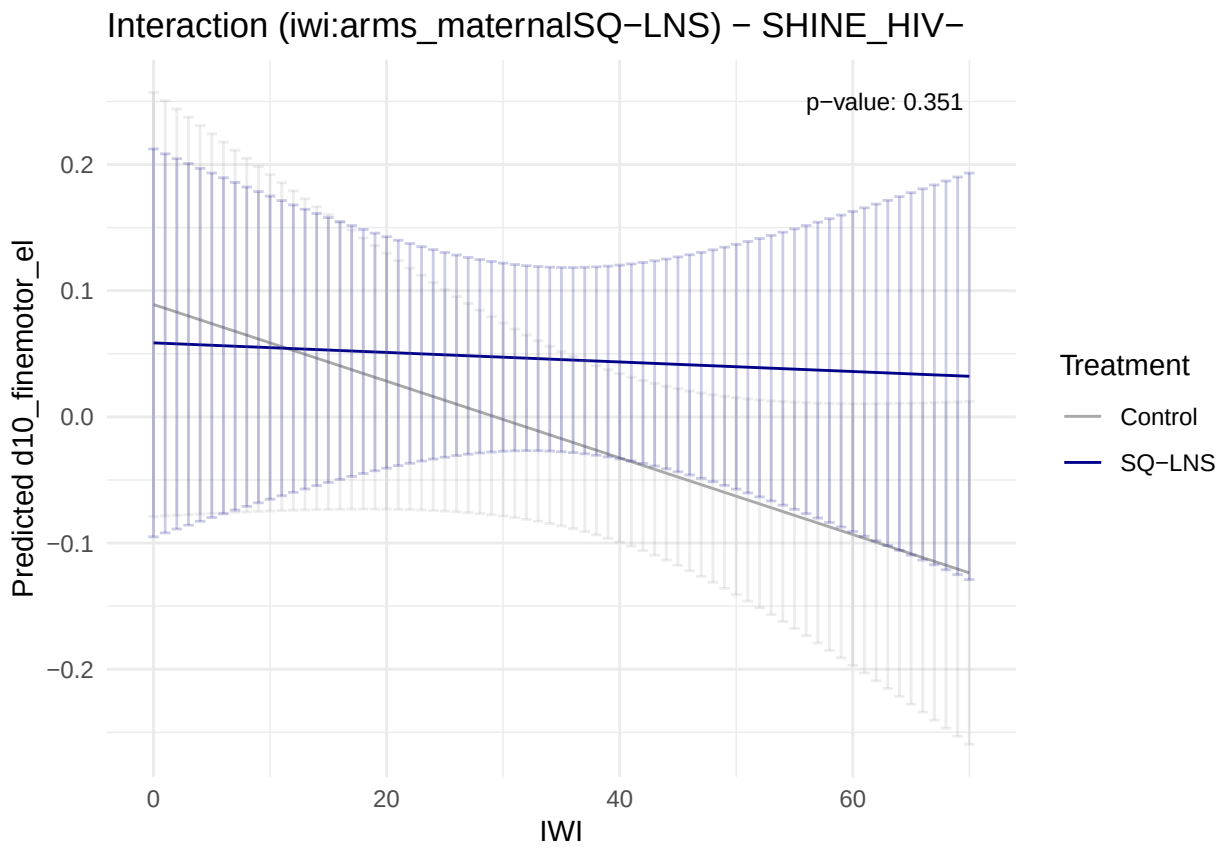
```
##
## $PROMISM_CS
```


Interaction (iwi:arms_maternalSQ−LNS) − PROMISM_CS
p−value: 0.415
Predicted d10_finemotor_el
0.8
0.4
0.0
−0.4
0
20
40
60
IWI
Treatment
Control
SQ−LNS

```
##
## $RDNS
```


Interaction (iwi:arms_maternalSQ−LNS) − RDNS
p−value: 0.871
Predicted d10_finemotor_el
0.3
0.0
−0.3
−0.6
0
20
40
60
IWI
Treatment
Control
SQ−LNS

```
##
## $`SHINE_HIV-`
```




```
point.wise.DF =  pointwise.ma(predictions,
                                        clustering.variable = "study",
                                        combining.variables = c("iwi","arms_maternal"),
                                        predicted.outcome =  "fit.fit",
                                        #predicted.outcome.se = "se.fit",
                                        predicted.outcome.CI = c("fit.lwr","fit.upr"),
                                        tau.method = "REML")
```

```
##
[1] "1%"
##
[1] "2%"
##
[1] "3%"
##
[1] "4%"
##
[1] "5%"
##
[1] "6%"
##
[1] "7%"
##
[1] "8%"
##
[1] "9%"
```

```
##
[1] "10%"
##
[1] "11%"
##
[1] "12%"
##
[1] "13%"
##
[1] "14%"
##
[1] "15%"
##
[1] "16%"
##
[1] "17%"
##
[1] "18%"
##
[1] "19%"
##
[1] "20%"
##
[1] "21%"
##
[1] "22%"
##
[1] "23%"
##
[1] "24%"
##
[1] "25%"
##
[1] "26%"
##
[1] "27%"
##
[1] "28%"
##
[1] "29%"
##
[1] "30%"
##
[1] "31%"
##
[1] "32%"
##
[1] "33%"
##
[1] "34%"
##
[1] "35%"
##
[1] "36%"
```

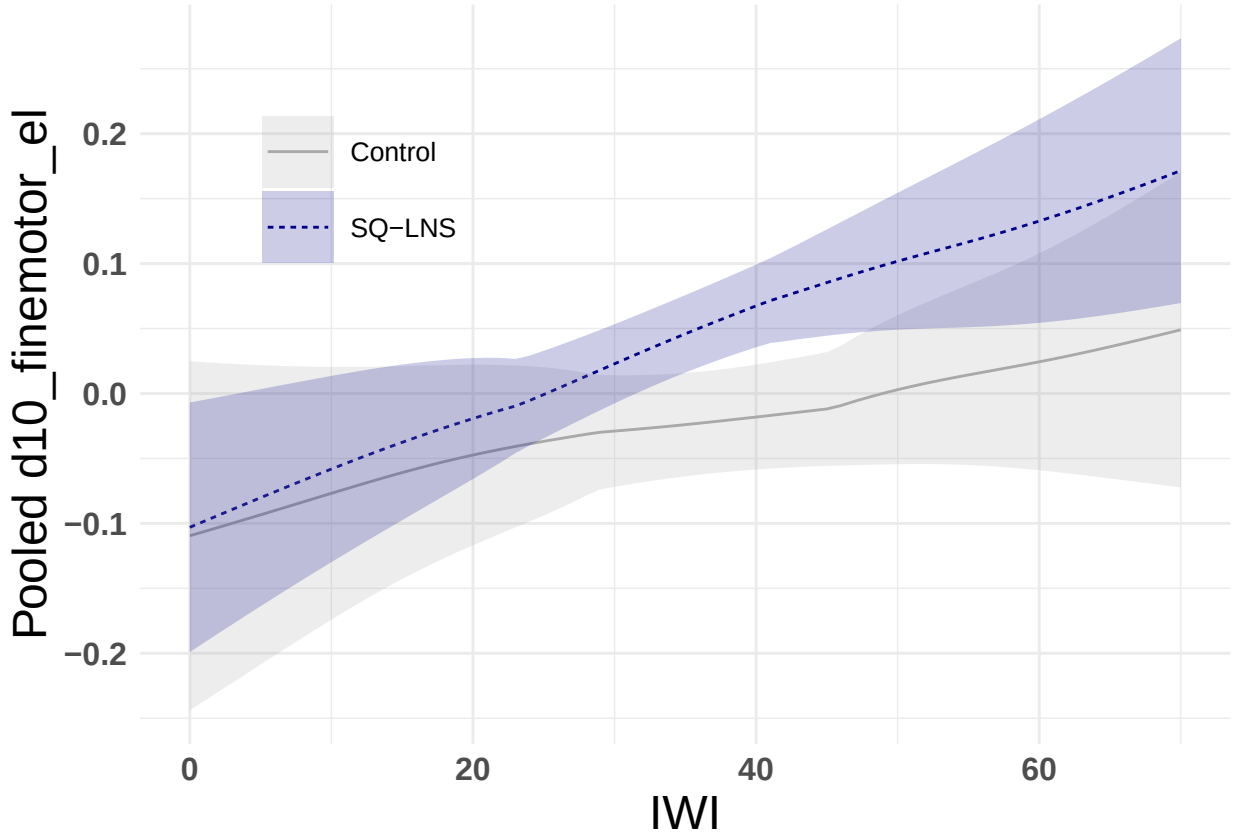
```
##
[1] "37%"
##
[1] "38%"
##
[1] "39%"
##
[1] "40%"
##
[1] "41%"
##
[1] "42%"
##
[1] "43%"
##
[1] "44%"
##
[1] "45%"
##
[1] "46%"
##
[1] "47%"
##
[1] "48%"
##
[1] "49%"
##
[1] "50%"
##
[1] "51%"
##
[1] "52%"
##
[1] "53%"
##
[1] "54%"
##
[1] "55%"
##
[1] "56%"
##
[1] "57%"
##
[1] "58%"
##
[1] "59%"
##
[1] "60%"
##
[1] "61%"
##
[1] "62%"
##
[1] "63%"
```

```
##
[1] "64%"
##
[1] "65%"
##
[1] "66%"
##
[1] "67%"
##
[1] "68%"
##
[1] "69%"
##
[1] "70%"
##
[1] "71%"
##
[1] "72%"
##
[1] "73%"
##
[1] "74%"
##
[1] "75%"
##
[1] "76%"
##
[1] "77%"
##
[1] "78%"
##
[1] "79%"
##
[1] "80%"
##
[1] "81%"
##
[1] "82%"
##
[1] "83%"
##
[1] "84%"
##
[1] "85%"
##
[1] "86%"
##
[1] "87%"
##
[1] "88%"
##
[1] "89%"
##
[1] "90%"
```

```
##
[1] "91%"
##
[1] "92%"
##
[1] "93%"
##
[1] "94%"
##
[1] "95%"
##
[1] "96%"
##
[1] "97%"
##
[1] "98%"
##
[1] "99%"
##
[1] "100%"
```

```
## Backtransform predicted outcomes and their corresponding confidence intervals
point.wise.DF$RE.meta  =        point.wise.DF$RE.meta
point.wise.DF$RE.meta.upper  =  point.wise.DF$RE.meta.upper
point.wise.DF$RE.meta.lower  =  point.wise.DF$RE.meta.lower
```

```
point.wise.DF.plot = point.wise.DF %>%
  mutate(Treatment = as.factor(arms_maternal)) %>%
  ggplot(aes(x = iwi, y = RE.meta, fill = Treatment, linetype = Treatment)) +
  geom_line(size=0.5, aes(colour = Treatment))+
  scale_color_manual(values = c('darkgrey','darkblue')) +
  scale_fill_manual(values = c('darkgrey','darkblue')) +
  geom_ribbon(aes(ymin = RE.meta.lower,ymax=RE.meta.upper),alpha=0.2) +
  #scale_color_jama(name= "Treatment")+
  #scale_linetype_discrete(name ="Treatment")+ylab("") +
  #scale_linetype_manual(values=c("solid","dashed")) +
  labs(color='Treatment', fill='Treatment', linetype='Treatment') +
  xlab(bquote('IWI')) +
  ylab(bquote('Pooled d10_finemotor_el')) +
  theme_minimal() +
  theme(legend.position = c(0.20,0.75),
        legend.title = element_blank(),
        axis.title.y = element_text(size = 18),
        axis.title.x = element_text(size = 18),
        strip.text = element_text(face="bold", size=14, hjust = 0.5),
        axis.text.y = element_text(face="bold",  size=12),
        axis.text.x = element_text(face="bold",  size=12),
        legend.key.size = unit(1, "cm"),
        legend.key.width = unit(1,"cm"),
        legend.text=element_text(size=10, hjust = 0))



print(point.wise.DF.plot)
```




```
predictions = predictions %>%
  mutate(iwi =  as.character(iwi))


absolute_diff = risk.diff.creator(dataframe = predictions,
                                  treatment = "arms_maternal", outcome = NULL,
                                  matching.variables = c("iwi","study"),
                                  predicted.outcome = "fit.fit",
                                  predicted.CI = c("fit.lwr","fit.upr"))


absolute_diff =  absolute_diff %>%
  select(study, iwi, fit.diff, diff.lower, diff.upper)
```

```
point.wise.absolute_diff  =  pointwise.ma(data = absolute_diff,
                                          clustering.variable = "study",
                                          combining.variables = c("iwi"),
                                          predicted.outcome = "fit.diff",
                                          predicted.outcome.se = NULL,
                                          predicted.outcome.CI = c("diff.lower","diff.upper"),
                                          tau.method = "REML"
)
```

```
##
[1] "1%"
```

```
##
[1] "3%"
##
[1] "4%"
##
[1] "6%"
##
[1] "7%"
##
[1] "8%"
##
[1] "10%"
##
[1] "11%"
##
[1] "13%"
##
[1] "14%"
##
[1] "15%"
##
[1] "17%"
##
[1] "18%"
##
[1] "20%"
##
[1] "21%"
##
[1] "23%"
##
[1] "24%"
##
[1] "25%"
##
[1] "27%"
##
[1] "28%"
##
[1] "30%"
##
[1] "31%"
##
[1] "32%"
##
[1] "34%"
##
[1] "35%"
##
[1] "37%"
##
[1] "38%"
##
[1] "39%"
```

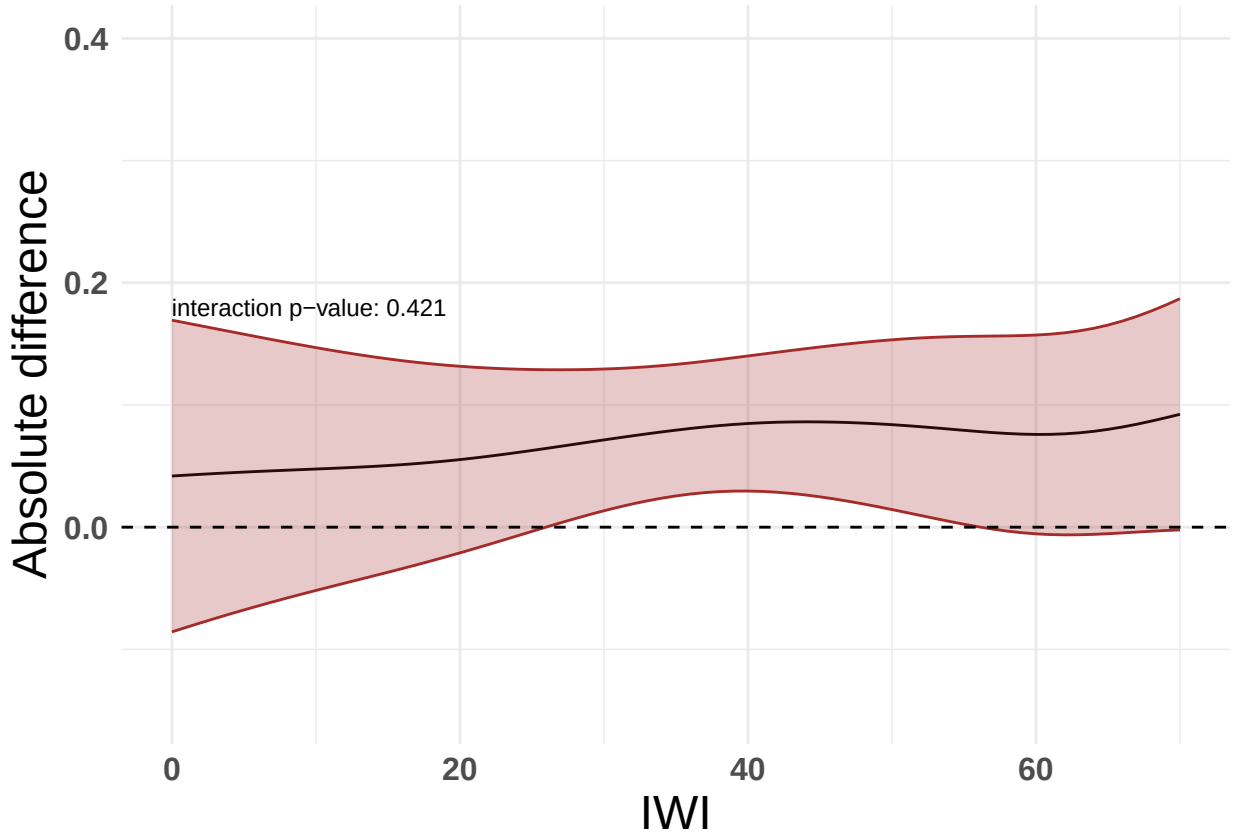
```
##
[1] "41%"
##
[1] "42%"
##
[1] "44%"
##
[1] "45%"
##
[1] "46%"
##
[1] "48%"
##
[1] "49%"
##
[1] "51%"
##
[1] "52%"
##
[1] "54%"
##
[1] "55%"
##
[1] "56%"
##
[1] "58%"
##
[1] "59%"
##
[1] "61%"
##
[1] "62%"
##
[1] "63%"
##
[1] "65%"
##
[1] "66%"
##
[1] "68%"
##
[1] "69%"
##
[1] "70%"
##
[1] "72%"
##
[1] "73%"
##
[1] "75%"
##
[1] "76%"
##
[1] "77%"
```

```
##
[1] "79%"
##
[1] "80%"
##
[1] "82%"
##
[1] "83%"
##
[1] "85%"
##
[1] "86%"
##
[1] "87%"
##
[1] "89%"
##
[1] "90%"
##
[1] "92%"
##
[1] "93%"
##
[1] "94%"
##
[1] "96%"
##
[1] "97%"
##
[1] "99%"
##
[1] "100%"
```

```
point.wise.absolute_diff =  point.wise.absolute_diff %>%
  mutate(iwi =  as.numeric(iwi))
```

```
point.wise.diff.plot = point.wise.absolute_diff %>%
  ggplot(aes(x = iwi,y=RE.meta)) + geom_line(size=0.5)+
  geom_ribbon(mapping = aes(ymin=RE.meta.lower, ymax=RE.meta.upper),alpha=0.25,
              color="brown", fill="brown")+
  geom_hline(yintercept = 0, linetype=2)+ylab("") +
  xlab(bquote('IWI')) +
  ylab(bquote('Absolute difference')) +
  theme_minimal() +
  theme(axis.title.y = element_text(size = 18),
        axis.title.x = element_text(size = 18),
        strip.text = element_text(face="bold", size=14, hjust = 0.5),
        axis.text.y = element_text(face="bold",  size=12),
        axis.text.x = element_text(face="bold",  size=12)) + ylim(c(-0.15,0.4)) +
      annotate("text", x = min(point.wise.absolute_diff$iwi) * 0.8,
               y = max(point.wise.absolute_diff$RE.meta.upper),
               label = paste("interaction p-value:",
                             round(as.numeric(unique(laz_eff_df_wide$pval.random)), 3)),
               hjust = 0, vjust = 1, size = 3, color = "black")
```

```
print(point.wise.diff.plot)
```



2nd method

1. Extract data for each study and fit the regression model.
2. Store the coefficients and variances from each model.
3. Combine the coefficients and variances into matrices for all studies.
4. Use random-effects meta-analysis to pool estimates for each coefficient, incorporating between-study variance.
5. Extract pooled estimates and variances for use in subsequent analyses.

```
study_list <- unique(df_analysis_fine$study)
results <- lapply(study_list, function(study) {
  df_study <- df_analysis_fine %>% filter(study == !!study)

  if (study %in% c("DOSE", "GHANA", "HAITI", "DYADG", "DYADM")) {
    model_study <- lm_robust(d10_finemotor_el ~ iwi + arms_maternal + iwi * arms_maternal,
                             data = df_study)
  } else {
    model_study <- lm_robust(d10_finemotor_el ~ iwi + arms_maternal + iwi * arms_maternal,
                             data = df_study, clusters = cluster)
  }
```

```
  list(
    coefficients = coef(model_study), # Extracts regression coefficients for each study.
    variances = diag(vcov(model_study)) # Extracts variances of the coefficients.
  )
})


# Extract coefficients and variances for meta-analysis
coefficients <- do.call(rbind, lapply(results, function(x) x$coefficients))
variances <- do.call(rbind, lapply(results, function(x) x$variances))
# do.call(rbind, ...) combines them into a matrix where rows represent studies and columns represent co
```

```
# Prepare pooled estimates
# The lapply loop performs meta-analysis for each regression coefficient (e.g., intercept, iwi, arms_ma
# Conducts random-effects meta-analysis for each coefficient, combining estimates across studies.
pooled_estimates <- lapply(1:ncol(coefficients), function(i) {
  metagen(
    TE = coefficients[, i],
    seTE = sqrt(unlist(variances)[, i]),
    studlab = study_list, # Labels for the studies.
    #sm = "MD",  # Mean difference
    method.tau = "REML", # REML method to estimate between-study variance.
    hakn = FALSE # Do not use Hartung-Knapp adjustment (alternative method for small sample sizes).
  )
})

# Extract pooled coefficients and variances
pooled_coefficients <- sapply(pooled_estimates, function(res) res$TE.random) # The pooled (random-effec
pooled_variances <- sapply(pooled_estimates, function(res) res$seTE.random^2) # The variance of the poo
names(pooled_coefficients) <- colnames(coefficients)
names(pooled_variances) <- colnames(variances)
```

Computes predicted values, standard errors, and pointwise 95% CIs for two different levels of the categorical variable arms_maternal (0 and 1) across a range of values of the continuous variable iwi

1. Compute the point estimates of the absolute difference.
2. Propagate the uncertainty (variance) from the main effect and interaction term, including their covariance.
3. Use the resulting variance to compute the standard error (SE) and construct confidence intervals for the absolute difference at each value of IWI.

```
# Define IWI values
iwi_vals <- seq(min(df_analysis_fine$iwi), by = 1, length.out = 71)
# length.out = 71 generates 71 evenly spaced values in this range including 0 (0-70). These values will

# Design matrices
X_0 <- cbind(1, iwi_vals, 0, 0)  # arms_maternal == 0
#cbind() creates a matrix for predicting outcomes when arms_maternal = 0.
#The columns of this matrix correspond to the terms in the regression model:
#1: Intercept (constant).
#iwi_vals: Values of the iwi predictor.
```

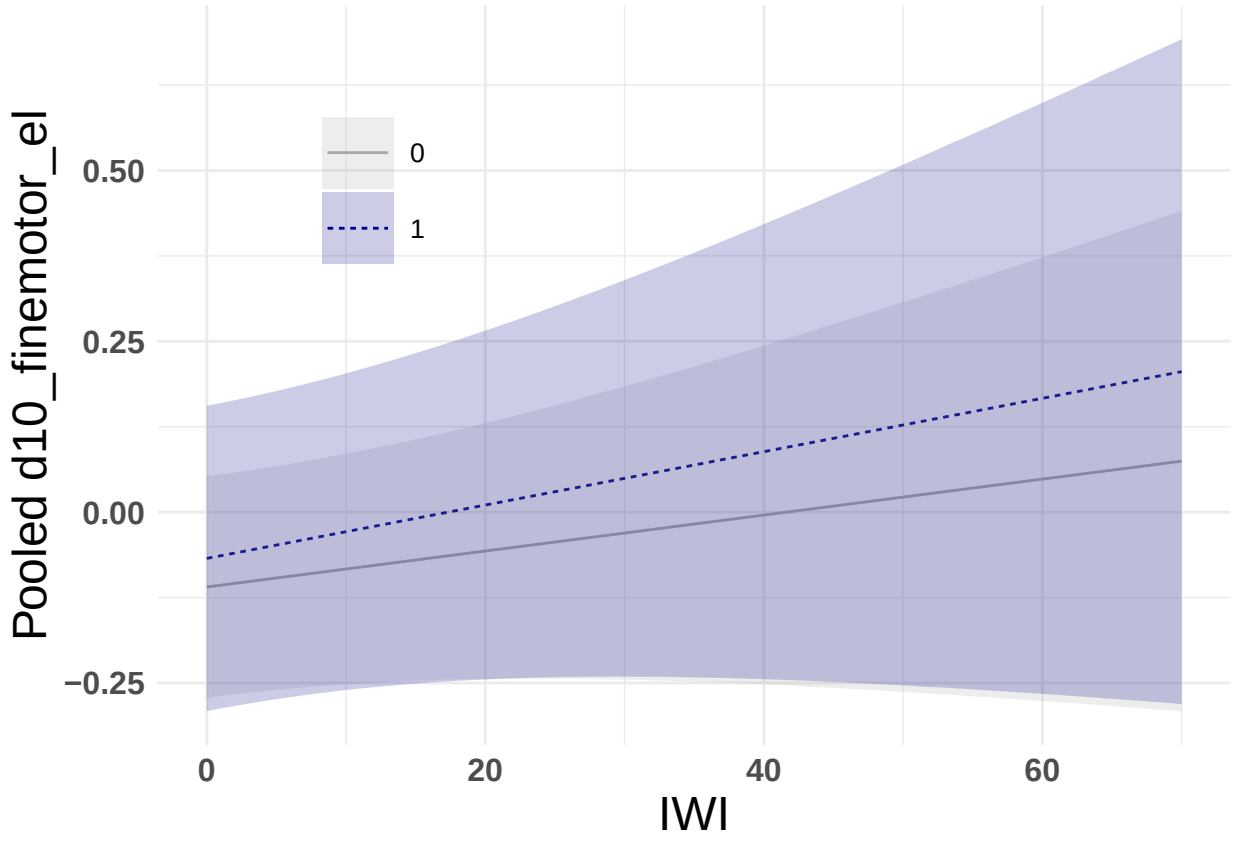
```r
#0: arms_maternal = 0 (since this is for the arms_maternal == 0 group).
#0: Interaction term iwi * arms_maternal is also 0 because arms_maternal = 0.
X_1 <- cbind(1, iwi_vals, 1, iwi_vals)  # arms_maternal == 1
#This matrix is for predicting outcomes when arms_maternal = 1.
#Columns correspond to:
#1: Intercept.
#iwi_vals: Same range of iwi values.
#1: arms_maternal = 1.
#iwi_vals: Interaction term iwi * arms_maternal, which equals iwi when arms_maternal = 1.


# Predictions
pred_0 <- X_0 %*% pooled_coefficients
pred_1 <- X_1 %*% pooled_coefficients
#coef(model) extracts the estimated regression coefficients from the lm_robust model.
#The %*% operator performs matrix multiplication:
#For X_0, predictions are calculated using the coefficients applied to the design matrix for arms_mater
#For X_1, predictions are calculated similarly for arms_maternal = 1.

# Variance-covariance matrix of pooled coefficients
pooled_vcov <- diag(pooled_variances)
#diag(pooled_variances) creates a diagonal variance-covariance matrix for the pooled coefficients.
#pooled_variances: Variances of the pooled coefficients (obtained from the meta-analysis).
#Diagonal Matrix: Assumes no covariances between the pooled coefficients (off-diagonal elements are zer
# Assign dimension names to pooled_vcov
dimnames(pooled_vcov) <- list(names(pooled_coefficients), names(pooled_coefficients))

# Variance for predictions
se_0 <- sqrt(rowSums((X_0 %*% pooled_vcov) * X_0))
#X_0: Design matrix for arms_maternal == 0 (defined previously):

#Each row of X_0 represents a combination of predictor values for a specific iwi.
#Columns correspond to the pooled coefficients (e.g., intercept, iwi, etc.).
#Matrix Multiplication:

#X_0 %*% pooled_vcov: Multiplies the design matrix (X_0) with the variance-covariance matrix (pooled_vc
#Element-Wise Multiplication:

#(X_0 %*% pooled_vcov) * X_0: Scales the variances and covariances by the values of the design matrix.
#Row-Wise Sum:

#rowSums(): Sums the variances and covariances for each prediction across rows, resulting in the total
#Square Root:

#sqrt(): Computes the standard errors (square root of the variances).

se_1 <- sqrt(rowSums((X_1 %*% pooled_vcov) * X_1))
# The same steps are followed, using X_1 (design matrix for arms_maternal == 1) instead of X_0.

# Pointwise 95% CIs
#The pointwise 95% CI is computed using the formula:

#CI=Prediction± t(0.975,df)×SE
```

```r
#qt(0.975, df = length(study_list) - 1)
#The critical value for the 95% confidence interval is derived from the t-distribution:
#0.975: 97.5th percentile of the t-distribution.
#df = length(study_list) - 1: Degrees of freedom for the meta-analysis, where length(study_list) is the

ci_lower_0 <- pred_0 - qt(0.975, df = length(study_list) - 1) * se_0
ci_upper_0 <- pred_0 + qt(0.975, df = length(study_list) - 1) * se_0
#Lower Bound: pred_0 - t-value * se_0.
#Upper Bound: pred_0 + t-value * se_0.
#The result is a pointwise 95% confidence interval for each iwi value when arms_maternal == 0.

ci_lower_1 <- pred_1 - qt(0.975, df = length(study_list) - 1) * se_1
ci_upper_1 <- pred_1 + qt(0.975, df = length(study_list) - 1) * se_1
#The same formula is applied to compute confidence intervals for arms_maternal == 1.
```

```r
# Absolute difference
delta_y <- pooled_coefficients["arms_maternalSQ-LNS"] + pooled_coefficients["iwi:arms_maternalSQ-LNS"] *
#delta_y represents the predicted difference in outcomes between the two groups (arms_maternal = 1 and
#This is derived from the pooled regression coefficients:
#arms_maternalSQ-LNS: The coefficient for the group effect (the difference at iwi = 0).
#iwi:arms_maternalSQ-LNS: The coefficient for the interaction term, which captures how the effect of ar

# Variance of the difference
var_delta_y <- pooled_variances["arms_maternalSQ-LNS"] +
  iwi_vals^2 * pooled_variances["iwi:arms_maternalSQ-LNS"] +
  2 * iwi_vals * pooled_vcov["arms_maternalSQ-LNS", "iwi:arms_maternalSQ-LNS"]
#The variance of the difference is calculated based on the variances and covariance of the pooled coeff
#iwi_vals^2 * pooled_variances["iwi:arms_maternalSQ-LNS"]:
#Contribution of the variance of the interaction term (iwi:arms_maternalSQ-LNS).
#Scaled by the square of iwi values since the interaction term is multiplied by iwi.
#2 * iwi_vals * pooled_vcov["arms_maternalSQ-LNS", "iwi:arms_maternalSQ-LNS"]:
#Contribution of the covariance between arms_maternalSQ-LNS and iwi:arms_maternalSQ-LNS.
#Accounts for how the two coefficients vary together.
#Scaled by 2 * iwi to reflect their interaction.

# Standard error and 95% CI
se_delta_y <- sqrt(var_delta_y)
ci_lower_delta <- delta_y - qt(0.975, df = length(study_list) - 1) * se_delta_y
ci_upper_delta <- delta_y + qt(0.975, df = length(study_list) - 1) * se_delta_y
```

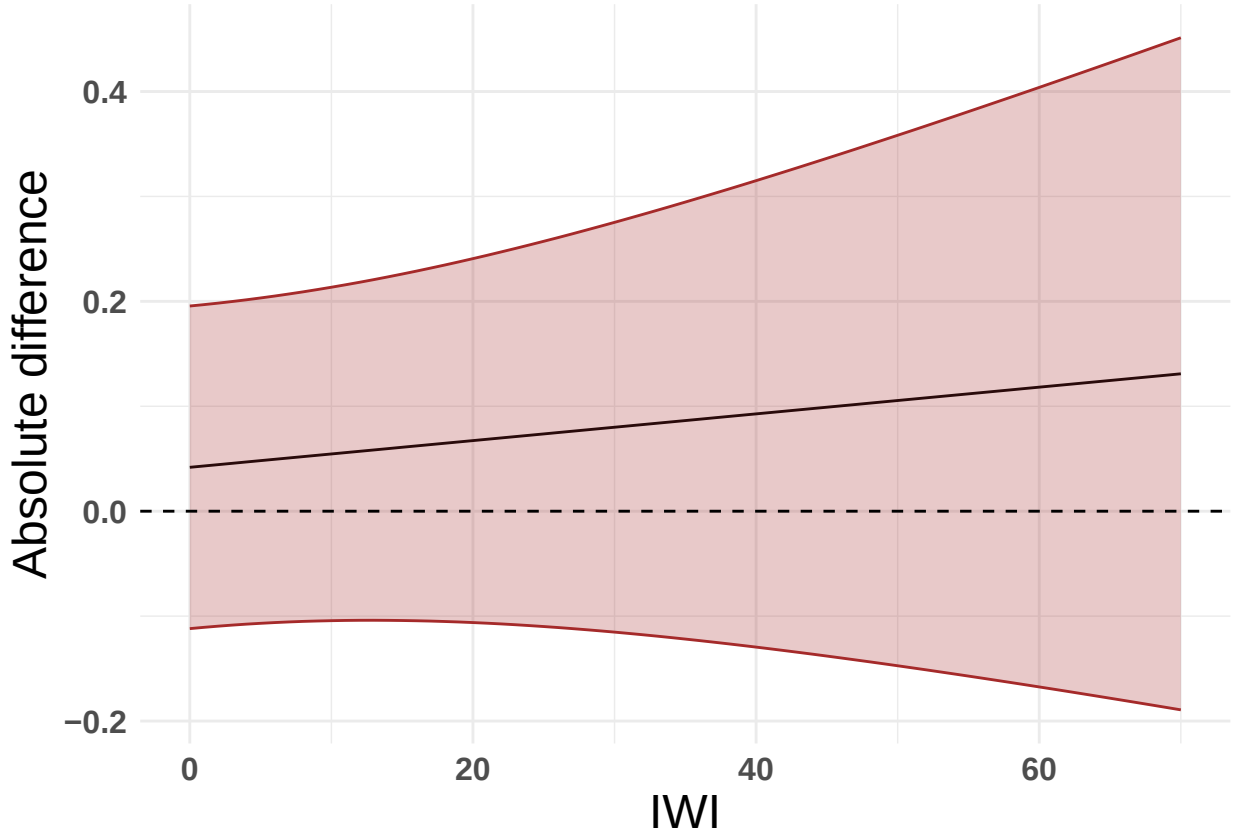
```r
# Create data for plotting
plot_fit <- data.frame(
  iwi = rep(iwi_vals, 2),
  predicted = c(pred_0, pred_1),
  ci_lower = c(ci_lower_0, ci_lower_1),
  ci_upper = c(ci_upper_0, ci_upper_1),
  arms_maternal = rep(c("0", "1"), each = length(iwi_vals))
)

# Plot
plot_fit %>%
  mutate(Treatment = as.factor(arms_maternal)) %>%
  ggplot(aes(x = iwi, y = predicted, fill = Treatment, linetype = Treatment)) +
```

```
  geom_line(size=0.5, aes(colour = Treatment)) +
  scale_color_manual(values = c('darkgrey','darkblue')) +
  scale_fill_manual(values = c('darkgrey','darkblue')) +
  geom_ribbon(aes(ymin = ci_lower, ymax = ci_upper, fill = arms_maternal), alpha = 0.2) +
  labs(color='Treatment', fill='Treatment', linetype='Treatment') +
  xlab(bquote('IWI')) +
  ylab(bquote('Pooled d10_finemotor_el')) +
  theme_minimal() +
  theme(legend.position = c(0.20,0.75),
        legend.title = element_blank(),
        axis.title.y = element_text(size = 18),
        axis.title.x = element_text(size = 18),
        strip.text = element_text(face="bold", size=14, hjust = 0.5),
        axis.text.y = element_text(face="bold",  size=12),
        axis.text.x = element_text(face="bold",  size=12),
        legend.key.size = unit(1, "cm"),
        legend.key.width = unit(1,"cm"),
        legend.text=element_text(size=10, hjust = 0))
```




```
# Create data for absolute difference
plot_diff <- data.frame(
  iwi = iwi_vals,
  delta_y = delta_y,
  ci_lower = ci_lower_delta,
  ci_upper = ci_upper_delta
```

```
)

# Plot
plot_diff %>%
  ggplot(aes(x = iwi, y = delta_y)) + geom_line(size=0.5)+
  geom_ribbon(aes(ymin = ci_lower, ymax = ci_upper), alpha = 0.25,
              color="brown", fill="brown") +
  geom_hline(yintercept = 0, linetype=2)+ylab("") +
  xlab(bquote('IWI')) +
  ylab(bquote('Absolute difference')) +
  theme_minimal() +
  theme(axis.title.y = element_text(size = 18),
        axis.title.x = element_text(size = 18),
        strip.text = element_text(face="bold", size=14, hjust = 0.5),
        axis.text.y = element_text(face="bold",  size=12),
        axis.text.x = element_text(face="bold",  size=12)) #+ ylim(c(-0.15,0.4)) +
```




```
      annotate("text", x = min(point.wise.absolute_diff$iwi) * 0.8,
               y = max(point.wise.absolute_diff$RE.meta.upper),
               label = paste("interaction p-value:",
                             round(as.numeric(unique(laz_eff_df_wide$pval.random)), 3)),
               hjust = 0, vjust = 1, size = 3, color = "black")
```

```
## mapping: x = ~x, y = ~y
## geom_text: na.rm = FALSE
```

```
## stat_identity: na.rm = FALSE
## position_identity
```

Session info

```
sessionInfo()
```

```
## R version 4.3.3 (2024-02-29)
## Platform: aarch64-apple-darwin20 (64-bit)
## Running under: macOS Sonoma 14.5
##
## Matrix products: default
## BLAS:   /System/Library/Frameworks/Accelerate.framework/Versions/A/Frameworks/vecLib.framework/Versio
## LAPACK: /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/lib/libRlapack.dylib;  LAPACK ve
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## time zone: America/Los_Angeles
## tzcode source: internal
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] geepack_1.3.10     estimatr_1.0.4     sandwich_3.1-0     msm_1.7.1
##  [5] lmtest_0.9-40      zoo_1.8-12         tidymv_3.4.2       meta_7.0-0
##  [9] metadat_1.2-0      mgcv_1.9-1         nlme_3.1-164       patchwork_1.2.0
## [13] RColorBrewer_1.1-3 cowplot_1.1.3      gridExtra_2.3      ggrepel_0.9.5
## [17] ggsci_3.0.1        broom_1.0.5        rmarkdown_2.26     skimr_2.1.5
## [21] readxl_1.4.3       lubridate_1.9.3    forcats_1.0.0      stringr_1.5.1
## [25] purrr_1.0.2        readr_2.1.5        tidyr_1.3.1        tibble_3.2.1
## [29] ggplot2_3.5.1      tidyverse_2.0.0    haven_2.5.4        table1_1.4.3
## [33] dplyr_1.1.4        here_1.0.1
##
## loaded via a namespace (and not attached):
##  [1] tidyselect_1.2.1    farver_2.1.2        fastmap_1.1.1       CompQuadForm_1.4.3
##  [5] mathjaxr_1.6-0      digest_0.6.35       timechange_0.3.0    lifecycle_1.0.4
##  [9] survival_3.5-8      magrittr_2.0.3      compiler_4.3.3      sass_0.4.9
## [13] rlang_1.1.4         tools_4.3.3         utf8_1.2.4          yaml_2.3.8
## [17] knitr_1.45          labeling_0.4.3      xml2_1.3.6          repr_1.1.7
## [21] expm_0.999-9        withr_3.0.0         numDeriv_2016.8-1.1 grid_4.3.3
## [25] fansi_1.0.6         colorspace_2.1-0    scales_1.3.0        MASS_7.3-60.0.1
## [29] tinytex_0.50        cli_3.6.2           mvtnorm_1.2-4       metafor_4.6-0
## [33] ragg_1.3.0          generics_0.1.3      rstudioapi_0.15.0   tzdb_0.4.0
## [37] cachem_1.0.8        minqa_1.2.6         splines_4.3.3       cellranger_1.1.0
## [41] base64enc_0.1-3     vctrs_0.6.5         boot_1.3-29         Matrix_1.6-5
## [45] jsonlite_1.8.8      hms_1.1.3           Formula_1.2-5       systemfonts_1.0.6
## [49] jquerylib_0.1.4     glue_1.7.0          nloptr_2.0.3        stringi_1.8.4
## [53] gtable_0.3.5        lme4_1.1-35.1       munsell_0.5.1       pillar_1.9.0
## [57] htmltools_0.5.7     R6_2.5.1            textshaping_0.3.7   rprojroot_2.0.4
```

```
## [61] evaluate_0.23       lattice_0.22-5      highr_0.10          backports_1.4.1
## [65] bslib_0.6.1         Rcpp_1.0.12         xfun_0.42           pkgconfig_2.0.3
```