Appendices

Appendix A: German scales used

Table 7: The German scales used to measure sustainability attitudes and sustainability behaviours based on the Theory of Planned Behaviour. Attitudes, subjective norms, perceived behavioural control and intentions were combined to form sustainability attitudes, and self-reported behaviours were taken as sustainability behaviours. The scales have been copied one to one from Pauli (2023).

#	Skala	Item	Theoretische Einordnung	Verwendete Frage auf Deutsch	Quelle
1	**Einstellungen zu klima-schützendem Verhalten**	AT1	Nachhaltigkeits-Einstellungen	Die Umwelt in Deutschland ist durch den globalen Klimawandel gefährdet.	Masud et al. 2016
2		AT2		Die derzeitige globale Erwärmung ist NICHT vom Menschen verursacht, sondern ein natürlicher Vorgang.	
3		AT3		Der Klimawandel schadet der natürlichen Umwelt und der Tierwelt in Deutschland.	
4		AT4		Ich bin bereit, einen gewissen Betrag zu bezahlen, um die Auswirkungen des Klimawandels zu verringern.	
5	**Subjektive Normen**	SN1		In meiner Familie wird oft über den Klimawandel oder die globale Erwärmung diskutiert.	Lin 2013
6		SN2		Meine Mitschülerinnen und Mitschüler diskutieren oft über den Klimawandel oder die globale Erwärmung.	
7		SN3		Meine Mitschülerinnen und Mitschüler könnten mich kritisieren, wenn ich keine Maßnahmen zum Klimaschutz ergreife.	
8	**Wahrgenommene Verhaltens-kontrolle**	PBC 1		Ich glaube, dass ich dazu beitragen kann, die Auswirkungen des Klimawandels abzuschwächen.	Pouya und Niyaz 2022
9		PBC 2		Durch mein Handeln im Alltag kann ich zur Verringerung von CO2-Ausstoß beitragen.	
10	**Intentionen**	IN1		Es liegt in meiner Verantwortung, meine Mitbürgerinnen und Mitbürger zu ermutigen, den Klimawandel zu beachten.	Pouya und Niyaz 2022
11		IN2		Ich bin bereit dazu, mich in meinem täglichen Leben umweltfreundlicher zu verhalten.	
12		IN3		Ich bin bereit, alles zu tun, um die Auswirkungen des Klimawandels zu mindern.	
13	**Verhalten**	B01	Nachhaltigkeits-Verhalten	Ich habe meinen Fleischkonsum in den letzten Monaten bewusst reduziert.	Lin 2013
14		B02		Ich kaufe in Deutschland produziertes Obst und vermeide den Kauf von importiertem Obst (z. B. Bananen, Kiwis).	
15		B03		In meinem Kühlschrank lagere ich oft Lebensmittel, die das Haltbarkeitsdatum überschritten haben.	
16		B04		Beim Kauf von Elektrogeräten achte ich am meisten auf den Preis der Geräte.	
17		B05		Ich kaufe Elektrogeräte, die ein Energiesparlabel haben.	
18		B06		Ich schalte Lichter und Wasserhähne so oft wie möglich aus.	
19		B07		Ich ziehe den Stecker von Geräten, die vorübergehend nicht in Gebrauch sind.	
20		B08		Ich fahre hauptsächlich mit einem Auto oder einem Motorroller, beziehungsweise werde gefahren.	
21		B09		Ich nutze Aufzüge und steige selten Treppen.	
22		B10		Ich unterstütze eine Erhöhung der Besteuerung von Kraftstoffen, um den Verbrauch fossiler Kraftstoffe zu reduzieren.	

Table 8: The German scales used for efficacy beliefs were based on the Triple-A framework (Agent-Action-Aim) by Hamann et al. (2024). The scales allow differentiation between personal and collective efficacy beliefs and between aim and action focus.

#	Skala	Theoretische Einstufung	Item	Verwendete Frage auf Deutsch	Quelle
1	Persönliche Selbstwirksamkeits-überzeugungen	Aim-focussed	SW01_01	Ich glaube, dass meine eigenen Handlungen einen Beitrag zum Klimaschutz leisten können, wenn ich das möchte.	Hamann et al. 2024
2		Aim-focussed	SW01_02	Ich glaube, dass ich den Klimaschutz vorantreiben kann, indem ich in meinem Umfeld über den Klimawandel aufkläre, wenn ich das möchte.	
3		Action-focussed	SW01_03_inverse	Ich glaube nicht, dass ich in der Lage bin, mich für den Klimaschutz einzusetzen.	
4		Aim-focussed	SW01_04	Ich glaube, dass ich dazu in der Lage bin, andere davon zu überzeugen, sich für mehr Klimaschutz einzusetzen, wenn ich das möchte.	
5		Aim-focussed	SW01_05_inverse	Ich glaube nicht, dass ich Möglichkeiten habe, einen Einfluss auf den Klimawandel zu nehmen.	
6		Action-focussed	SW01_06	Ich glaube, dass ich beeinflussen kann, wie meine Schulleitung oder meine Schule bezogen auf den Klimaschutz handelt, wenn ich das möchte.	
7		Aim-focussed	SW01_07	Ich glaube, dass ich meine Schulleitung oder Schule dabei unterstützen kann, sich für Klimaschutz einzusetzen, wenn ich das möchte.	
8		Aim-focussed	SW01_08	Ich glaube, dass ich mich in Zusammenarbeit mit anderen sinnvoll für den Klimaschutz engagieren kann, wenn ich das möchte.	
9	Kollektive Wirksamkeits-überzeugungen	Aim-focussed	CS01_01	Wir, als Schülerinnen und Schüler, können durch unsere Handlungen einen Beitrag zum Kllimaschutz leisten, wenn wir das möchten.	
10		Action-focussed	CS01_02	Wir, als Schülerinnen und Schüler, können beeinflussen, wie unsere Schulleitung oder Schule bezogen auf den Klimaschutz handelt, wenn wir das möchten.	
11		Aim-focussed	CS01_03	Wir, als Schülerinnen und Schüler, sind in der Lage, andere davon zu überzeugen, sich für mehr Klimaschutz einzusetzen, wenn wir das möchten.	

Appendix B: Form Declaration of consent by parents

Einverständniserklärung der Eltern

Teilnahme an einer Untersuchung zum Thema „Nachhaltigkeitskompetenzen im Projekt KlimaRatSchule“

Liebe Eltern,

im Rahmen meiner **Masterarbeit** an der Universität Freiburg im Studiengang „Environmental Governance“ begleite ich, Daniela Gargya, das **Projekt „KlimaRatSchule“** (KRS), welches letztes Jahr an der Schule durchgeführt wurde. Mein Ziel ist es nun die längerfristige Wirkung des Projekts, über das Ende hinausgehend, zu untersuchen.

Inwiefern Schülerinnen und Schüler, welche an dem Projekt teilgenommen haben, veränderte Einstellungen und Selbstwirksamkeitsüberzeugungen, also die Überzeugung Vorgänge verstehen und beeinflussen zu können, aufweisen ist ein wesentlicher Teil der Arbeit.

Hierzu füllen die Schülerinnen und Schüler, welche am Projekt teilgenommen haben, sowie eine „Kontrollgruppe“, einen identischen Fragebogen online aus. In diesem werden Fragen zu Einstellungen, Verhalten und Selbstwirksamkeit abgefragt. So kann der Einfluss des Projektes auf Nachhaltigkeitskompetenzen untersucht werden.

Das Ausfüllen des Fragebogens erfordert ca. 10-15 Minuten und erfolgt in Anwesenheit einer Lehrperson.

Selbstverständlich erfolgt **eine Anonymisierung aller personenbezogenen Daten**, sodass keine Ergebnisse auf einzelne Personen rückführbar sind.

Falls Sie noch Fragen haben, können Sie sich sehr gerne mit mir in Verbindung setzen. Ich danke Ihnen für Ihre Unterstützung.

Mit freundliche Grüßen,

Daniela Gargya

Universität Freiburg
E-Mail: daniela@gargya.de

Albert-Ludwigs-Universität Freiburg

Fakultät für Umwelt und natürliche Ressourcen

Teilnahme an einer Untersuchung zum Thema
„Nachhaltigkeitskompetenzen im Projekt KlimaRatSchule“

Einverständniserklärung der Eltern

Mir ist das Ziel und Wesen der Forschungsprojektes bewusst. Ich hatte die Möglichkeit Fragen zu stellen und habe zurzeit keine weiteren Fragen mehr.

Ich weiß, dass die Teilnahme meines Kindes daran freiwillig ist. Ich weiß, dass ich jederzeit und ohne Angabe von Gründen diese Zustimmung widerrufen kann, ohne dass sich dieser Entschluss nachteilig auf mein Kind auswirken wird. Ich bin damit einverstanden, dass in diesem Forschungsprojekt Daten von meinem Kind in Form von Befragungen erhoben werden. Mir ist bekannt, dass die erhobenen Daten anonym gespeichert und ausschließlich für wissenschaftliche Zwecke verwendet werden.

Hiermit erkläre ich mich einverstanden, dass

Meine Tochter/mein Sohn _____

freiwillig an einer Untersuchung zum Thema „Nachhaltigkeitskompetenzen im Projekt KlimaRatSchule“ teilnimmt.

_____ _____

Ort, Datum Unterschrift Erziehungsberechtigte:r

Appendix C: Information slide given to students at measurement point 3

Appendix D: Statistical test results

Table 9: Kruskal-Wallis test results comparing scores across 3 measurement points within groups (RQ1).

Scale	Group	P-Value	Kruskal-Wallis H
AT_Mean	Control group	0.7653188	0.5349257
AT_Mean	Involved group	0.3578105	2.0555033
B_Mean	Control group	0.7167712	0.6659973
B_Mean	Involved group	0.2900163	2.4756363
INT_Mean	Control group	0.1033114	4.5400142
INT_Mean	Involved group	0.9685065	0.0640002
PBC_Mean	Control group	0.4085675	1.7901962
PBC_Mean	Involved group	0.0862698	4.9005516
SN_Mean	Control group	0.7083967	0.6895022
SN_Mean	Involved group	0.6809252	0.7686057
TPB_Mean	Control group	0.2351661	2.8949266
TPB_Mean	Involved group	0.1762751	3.4714192

Table 10: Wilcoxon test results comparing groups (RQ1).

Scale	Measurement Point	P-Value	W Statistic	Significance symbols
AT_Mean	Measurement point 1	0.1203383	429.0	
AT_Mean	Measurement point 2	0.0024719	302.5	**
AT_Mean	Measurement point 3	0.0884757	87.5	
B_Mean	Measurement point 1	0.0021725	300.0	**

Scale	Measurement Point	P-Value	W Statistic	Significance symbols
B_Mean	Measurement point 2	0.0000515	210.5	***
B_Mean	Measurement point 3	0.0319491	71.5	*
INT_Mean	Measurement point 1	0.0009521	283.0	***
INT_Mean	Measurement point 2	0.0000551	214.5	***
INT_Mean	Measurement point 3	0.0145942	62.0	*
PBC_Mean	Measurement point 1	0.1694962	446.0	
PBC_Mean	Measurement point 2	0.0005760	270.5	***
PBC_Mean	Measurement point 3	0.2163233	104.5	
SN_Mean	Measurement point 1	0.1392966	435.5	
SN_Mean	Measurement point 2	0.0041714	317.0	**
SN_Mean	Measurement point 3	0.1142683	92.0	
TPB_Mean	Measurement point 1	0.0043080	317.5	**
TPB_Mean	Measurement point 2	0.0000257	196.0	***
TPB_Mean	Measurement point 3	0.0082193	54.0	**

Table 11: Wilcoxon test results comparing personal/collective efficacy within groups (RQ3).

Group	W Statistic	P-Value
Control group	1	1
Involved group	0	1

Table 12: Wilcoxon test results comparing personal/collective efficacy between groups (RQ3).

Scale	P-Value	W Statistic
CS_Mean	0.6103249	123.0
SW_Mean	0.0968496	88.5

Table 13: Wilcoxon test results comparing aim/action focused efficacy within groups (RQ3).

Group	W Statistic	P-Value
Control group	1	1
Involved group	1	1

Table 14: Wilcoxon test results comparing aim/action focused efficacy between groups (RQ3).

Theoretical classification	P-Value	W Statistic	Significance symbol
action	0.2000000	1	
aim	0.0135185	8	*

Appendix E: R Code

R Code, including outputs, can be accessed through GitHub https://github.com/DaniGargya/MEG_thesis.git.

Data formatting and cleaning

```
# Full R script analysing sustainability competencies of students
# Master thesis project 2024, University of Freiburg
# Daniela Gargya
# July 2024

# loading libraries ----
library(tidyverse)
library(ggplot2)
library(dplyr)
library(haven) # to transform data from sav to csv
library(readr) # to transform data from sav to csv

# data conversion Paulis data from sav to csv ----
#pre angell
lisa_data_angell_pre <- read_sav("data/data_collection/data_pauli/ANGELL_PRE_anonym.SAV")
write_csv(x=lisa_data_angell_pre, path="data/data_collection/data_pauli/angell_pre.csv")

#post angell
lisa_data_angell_post <- read_sav("data/data_collection/data_pauli/ANGELL_POST_anonym.SAV")
write_csv(x=lisa_data_angell_post, path="data/data_collection/data_pauli/angell_post.csv")


# import MP3 data and formating ----
## copied from scoscie website: GNU R-SCript für Daten-Import
ds_file = "rdata_KRSumfrage_2024-06-07_10-21.csv"
options(encoding = "UTF-8")
ds = read.delim(
  file=ds_file, encoding="UTF-8", fileEncoding="UTF-8",
  header = FALSE, sep = "\t", quote = "\"",
  dec = ".", row.names = NULL,
  col.names = c(
    "CASE","SERIAL","REF","QUESTNNR","MODE","STARTED","AT01_01","AT01_02","AT01_03",
    "AT01_04","B001_01","B001_02","B001_03","B001_04","B001_05","B001_06","B001_07",
    "B001_08","B001_09","B001_10","CS01_01","CS01_02","CS01_03","IN01_01","IN01_02",
    "IN01_03","PB01_01","PB01_02","SN01_01","SN01_02","SN01_03","SW01_01","SW01_02",
    "SW01_03","SW01_04","SW01_05","SW01_06","SW01_07","SW01_08","WD01","WD02",
    "WD02_01","WD02_02","WD02_03","TIME001","TIME002","TIME003","TIME004","TIME005",
    "TIME006","TIME007","TIME008","TIME009","TIME_SUM","MAILSENT","LASTDATA",
    "FINISHED","Q_VIEWER","LASTPAGE","MAXPAGE","MISSING","MISSREL","TIME_RSI"
  ),
  as.is = TRUE,
  colClasses = c(
    CASE="numeric", SERIAL="character", REF="character", QUESTNNR="character",
    MODE="factor", STARTED="character", AT01_01="numeric", AT01_02="numeric",
    AT01_03="numeric", AT01_04="numeric", B001_01="numeric", B001_02="numeric",
    B001_03="numeric", B001_04="numeric", B001_05="numeric", B001_06="numeric",
    B001_07="numeric", B001_08="numeric", B001_09="numeric", B001_10="numeric",
    CS01_01="numeric", CS01_02="numeric", CS01_03="numeric", IN01_01="numeric",
    IN01_02="numeric", IN01_03="numeric", PB01_01="numeric", PB01_02="numeric",
    SN01_01="numeric", SN01_02="numeric", SN01_03="numeric", SW01_01="numeric",
    SW01_02="numeric", SW01_03="numeric", SW01_04="numeric", SW01_05="numeric",
    SW01_06="numeric", SW01_07="numeric", SW01_08="numeric", WD01="numeric",
```

```
    WD02="numeric", WD02_01="logical", WD02_02="logical", WD02_03="logical",
    TIME001="integer", TIME002="integer", TIME003="integer", TIME004="integer",
    TIME005="integer", TIME006="integer", TIME007="integer", TIME008="integer",
    TIME009="integer", TIME_SUM="integer", MAILSENT="character",
    LASTDATA="character", FINISHED="logical", Q_VIEWER="logical",
    LASTPAGE="numeric", MAXPAGE="numeric", MISSING="numeric", MISSREL="numeric",
    TIME_RSI="numeric"
  ),
  skip = 1,
  check.names = TRUE, fill = TRUE,
  strip.white = FALSE, blank.lines.skip = TRUE,
  comment.char = "",
  na.strings = ""
)

row.names(ds) = ds$CASE

rm(ds_file)


### assign treatment/ control groups ----
ds <- ds %>%
  mutate(Group = ifelse(WD02_01, "group2", ifelse(WD02_03, "group0", "group1")))


# exclude incomplete data ----
# Function to count the number of NAs in a row
count_nas <- function(row) {
  sum(is.na(row))
}

# Apply the function to each row and create a new column 'count_nas'
ds$count_nas <- apply(ds, 1, count_nas)

# Filter the data frame to exclude rows with NAs
ds_filtered <- ds[ds$count_nas <= 1, ]


# exclude data with more than 25% missing (8 answers) ----
# Function to count the number of -1s in a row
count_minus_ones <- function(row) {
  sum(row == -1, na.rm = TRUE)
}

ds_filtered$count_minus_ones <- apply(ds_filtered, 1, count_minus_ones)

# Filter the data frame to exclude rows with more than eight -1s
ds_filtered <- ds_filtered[ds_filtered$count_minus_ones <= 8, ]

# exclude data with less than  2.5 min/ 150sec! processing time (instead of 4)----
ds_filtered <- ds_filtered[ds_filtered$TIME_SUM >= 150, ]
```

```r
# data transformation # scaling according to Pauli (0-3, instead of 1-4) ----
# Define a function to convert values
convert_values <- function(x) {
  x <- ifelse(x == 4, 3,
              ifelse(x == 3, 2,
                     ifelse(x == 2, 1,
                            ifelse(x == 1, 0,
                                   ifelse(x == -1, -100,
                                          x)))))
  return(x)
}


# Apply the function to all columns of the data frame
ds_scaled <- lapply(ds_filtered, convert_values)
ds_scaled <- as.data.frame(ds_scaled)

# invert certain scales to reflect meaning (AT2, B3, B4, B8, B9, SW3, SW5) ----
# Define a function to inverse scales and add new columns
inverse_scale_and_add_columns <- function(df, columns) {
  for (col in columns) {
    new_col_name <- paste(col, "inverse", sep = "_")
    df[[new_col_name]] <- inverse_scale(df[[col]])
  }
  return(df)
}

# Apply the function to add extra columns with inverse scales
ds_scaled_in <- inverse_scale_and_add_columns(ds_scaled, c("AT01_02", "B001_03", "B001_04", "B001_08",

# save formated dataset as csv (angell_mzp3) ----
write_csv(x=ds_scaled_in, path="data/data_collection/angell_mzp3.csv")
```

Data analysis and visualisation

```r
# Data analysis and visualisation

# loading libraries ----
library(tidyverse)
library(ggplot2)
library(RColorBrewer)
library(ggthemes) # for data visualisation
library(treemapify) # for data vis
library(dplyr)
library(purrr)
library(psych) # for cronbach alpha
library(ggrepel)
library(readxl)
library(knitr)

# create colour palette ----
display.brewer.pal(n = 8, name = 'Dark2')
```

```r
brewer.pal(n = 8, name = "Dark2")

# clean theme for graphs ----
theme_clean <- function(){
  theme_bw() +
    theme(axis.text.x = element_text(size = 14),
          axis.text.y = element_text(size = 14),
          axis.title.x = element_text(size = 14, face = "plain"),
          axis.title.y = element_text(size = 14, face = "plain"),
          panel.grid.major.x = element_blank(),
          panel.grid.minor.x = element_blank(),
          panel.grid.minor.y = element_blank(),
          panel.grid.major.y = element_blank(),
          plot.margin = unit(c(0.5, 0.5, 0.5, 0.5), units = , "cm"),
          plot.title = element_text(size = 15, vjust = 1, hjust = 0.5),
          legend.text = element_text(size = 12, face = "italic"),
          legend.title = element_text(size = 12, face = "bold"),
          legend.position = c(0.2, 0.8))
}

# importing relevant data ----
mzp1_clean <- read.csv("data/data_collection/data_pauli/angell_pre.csv")
mzp2_clean <- read.csv("data/data_collection/data_pauli/angell_post.csv")
mzp3_clean <- read.csv("data/data_collection/angell_mzp3.csv")


### calculate means MP3 ----
# Define a function to calculate the mean, excluding -100 and NAs
mean_exclude_negative_100 <- function(row) {
  values <- row[row != -100 & !is.na(row)]
  if (length(values) == 0) {
    return(NA)  # Return NA if all values are excluded
  }
  return(mean(values))
}

# Calculate the mean values for specific groups of columns and add them as new columns
#AT_mean
mzp3_clean$AT_mean <- apply(mzp3_clean[, c("AT01_01", "AT01_02_inverse", "AT01_03", "AT01_04")], 1, mea

#B_mean
mzp3_clean$B_mean <- apply(mzp3_clean[, c("B001_01", "B001_02", "B001_03_inverse", "B001_04_inverse", "

# CS_mean
mzp3_clean$CS_mean <- apply(mzp3_clean[, c("CS01_01", "CS01_02", "CS01_03")], 1, mean_exclude_negative_

# IN_mean
mzp3_clean$IN_mean <- apply(mzp3_clean[, c("IN01_01", "IN01_02", "IN01_03")], 1, mean_exclude_negative_

# PB_mean
mzp3_clean$PB_mean <- apply(mzp3_clean[, c("PB01_01", "PB01_02")], 1, mean_exclude_negative_100)

# SN_mean
```

```r
mzp3_clean$SN_mean <- apply(mzp3_clean[, c("SN01_01", "SN01_02", "SN01_03")], 1, mean_exclude_negative_

# SW_mean
mzp3_clean$SW_mean <- apply(mzp3_clean[, c("SW01_01", "SW01_02", "SW01_03_inverse", "SW01_04", "SW01_05

# TPB_mean
mzp3_clean$TPB_mean <- apply(mzp3_clean[, c("AT_mean", "SN_mean", "PB_mean", "IN_mean", "B_mean")], 1, m

# SW_CS_mean
mzp3_clean$SW_CS_mean <- apply(mzp3_clean[, c("SW_mean", "CS_mean")], 1, mean_exclude_negative_100)


### harmonising and combining dfs ----
# Add a time point indicator to each data frame
mzp1_clean$TP <- "Measurement point 1"
mzp2_clean$TP <- "Measurement point 2"
mzp3_clean$TP <- "Measurement point 3"


# MZP3
# Select only the columns that include '_mean' in their names along with 'TP' and 'Group'
selected_columns_t3 <- mzp3_clean %>%
  select(TP, Group, contains("_mean")) %>%
  rename (Time_Point = TP)


# Convert to long format
long_df_t3 <- selected_columns_t3 %>%
  pivot_longer(
    cols = contains("_mean"),
    names_to = "Category",
    values_to = "MeanValue")%>%
  mutate(Category = case_when(
    Category == "AT_mean" ~ "AT_Mean",
    Category == "B_mean" ~ "B_Mean",
    Category == "CS_mean" ~ "CS_Mean",
    Category == "IN_mean" ~ "INT_Mean",
    Category == "PB_mean" ~ "PBC_Mean",
    Category == "SN_mean" ~ "SN_Mean",
    Category == "SW_mean" ~ "SW_Mean",
    Category == "TPB_mean" ~ "TPB_Mean",
    Category == "SW_CS_mean" ~ "SW_CS_Mean",
    TRUE ~ Category  # Keep other values unchanged
  ))

# mzp2
# Select only the columns that include '_Mean' in their names along with 'TP' and 'Group'
selected_columns_t2 <- mzp2_clean %>%
  select(TP, Gruppe, contains("_Mean")) %>%
  rename(Time_Point = TP,
         Group = Gruppe) %>%
  mutate(Group = case_when(
   Group == 0 ~ "group0",
```

```
    Group == 1 ~ "group1",
   Group == 2 ~ "group2",
   TRUE ~ as.character(Group))) %>%
  mutate(Group = as.factor(Group))

# Convert to long format
long_df_t2 <- selected_columns_t2 %>%
  pivot_longer(
    cols = contains("_Mean"),
    names_to = "Category",
    values_to = "MeanValue"
  )


# mzp 1
# Select only the columns that include '_mean' in their names along with 'TP' and 'Group'
selected_columns_t1 <- mzp1_clean %>%
  rename(TPB_Mean = TPB_Mittelwert,
         Group = Gruppe,
         Time_Point = TP) %>%
 select(Time_Point, Group, contains("_Mean")) %>%
 mutate(Group = case_when(
  Group == 0 ~ "group0",
  Group == 1 ~ "group1",
  Group == 2 ~ "group2",
  TRUE ~ as.character(Group))) %>%
 mutate(Group = as.factor(Group))

# Convert to long format
long_df_t1 <- selected_columns_t1 %>%
    pivot_longer(
      cols = contains("_Mean"),
      names_to = "Category",
      values_to = "MeanValue"
    )

# Combine the data frames
combined_df <- rbind(long_df_t1, long_df_t2, long_df_t3) #1617 obs.

# Convert Group, Category and TP to factors
combined_df$Group <- as.factor(combined_df$Group)
combined_df$Time_Point <- as.factor(combined_df$Time_Point)
combined_df$Category <- as.factor(combined_df$Category)

### excluding group 1 from dataset ----
combined_df_g02 <- combined_df %>%
  filter(Group != "group1")

### check for normality of data ----
### MZP1
# exclude group1
selected_columns_t1 <- selected_columns_t1 %>%
  filter(Group != "group1")
```

```r
# Find columns with '_Mean' in their name
mean_cols_t1 <- grep("_Mean", names(selected_columns_t1), value = TRUE)

# Function to conduct normality test and return results
normality_test_t1 <- function(col) {
  shapiro_test <- shapiro.test(selected_columns_t1[[col]])
  tibble(Column = col,
         Test = "Shapiro-Wilk",
         W = shapiro_test$statistic,
         P_Value = shapiro_test$p.value,
         Normal = shapiro_test$p.value >= 0.05)
}

# Map function over columns and combine results
normality_results_t1 <- map_dfr(mean_cols_t1, normality_test_t1)

### MZP2
# exclude group1
selected_columns_t2 <- selected_columns_t2 %>%
  filter(Group != "group1")

# Find columns with '_Mean' in their name
mean_cols_t2 <- grep("_Mean", names(selected_columns_t2), value = TRUE)

# Function to conduct normality test and return results
normality_test_t2 <- function(col) {
  shapiro_test <- shapiro.test(selected_columns_t2[[col]])
  tibble(Column = col,
         Test = "Shapiro-Wilk",
         W = shapiro_test$statistic,
         P_Value = shapiro_test$p.value,
         Normal = shapiro_test$p.value >= 0.05)
}

# Map function over columns and combine results
normality_results_t2 <- map_dfr(mean_cols_t2, normality_test_t2)

### MZP3
# exclude group1
selected_columns_t3 <- selected_columns_t3 %>%
  filter(Group != "group1")

# Find columns with '_Mean' in their name
mean_cols_t3 <- grep("_mean", names(selected_columns_t3), value = TRUE)

# Function to conduct normality test and return results
normality_test_t3 <- function(col) {
  shapiro_test <- shapiro.test(selected_columns_t3[[col]])
  tibble(Column = col,
         Test = "Shapiro-Wilk",
         W = shapiro_test$statistic,
         P_Value = shapiro_test$p.value,
         Normal = shapiro_test$p.value >= 0.05)
```

```
}

# Map function over columns and combine results
normality_results_t3 <- map_dfr(mean_cols_t3, normality_test_t3)


### calculating internal validity with cronbach alpha MZP3 ----
# Convert -100 to NA
mzp3_cleaner <- mzp3_clean %>%
  filter(Group != "group1") %>%
  mutate(across(everything(), ~ ifelse(. == -100, NA, .)))

# Get column names with _inverse
columns_with_inverse <- grep("_inverse$", names(mzp3_cleaner), value = TRUE)
# Remove "_inverse" from column names
columns_to_exclude_without_inverse <- gsub("_inverse", "", columns_with_inverse)

filtered_df_cronbach <- mzp3_cleaner %>%
  select(matches("_0[1-9]$|_10$|_inverse"), -matches("WD")) %>%
  select(-one_of(columns_to_exclude_without_inverse)) %>%
  select(-c("B001_03_inverse", "B001_04_inverse")) #excluding these two items to  increase Cronbachs al

# handle missing values by using the mean
filtered_df_cronbach <- filtered_df_cronbach %>%
  mutate(across(everything(), ~ ifelse(is.na(.), mean(., na.rm = TRUE), .)))

# Group the columns based on the first two letters of the column name
groups <- split.default(filtered_df_cronbach, sub("^(..).*", "\\1", names(filtered_df_cronbach)))

# Calculate Cronbach's alpha for each group
alpha_results <- lapply(groups, function(x) {
  result <- alpha(x)
  result$total$raw_alpha  # Extracting raw alpha from the result
})


# Create a dataframe with scale names and Cronbach's alpha values
df_alpha <- data.frame(
  Scale = names(alpha_results),
  Alpha = unlist(alpha_results)
)


# Statistical analyses ----
### RQ1 Comparing SA/SB over time and between groups ----
# Kruskal-Wallis test for time points within each category and group
kw_results <- combined_df_g02 %>%
  filter(!Category %in% c("SW_Mean", "CS_Mean", "SW_CS_Mean")) %>% #exclude irrelevant categories for t
  group_by(Category, Group) %>%
  summarise(
    P_value = kruskal.test(MeanValue ~ Time_Point)$p.value,
    Kruskal_Wallis_H = kruskal.test(MeanValue ~ Time_Point)$statistic,  # Extract Kruskal-Wallis H stat
    .groups = 'drop') #%>%
```

```
  mutate(significance_time = case_when(
      kruskal_p_time < 0.001 ~ "***",
      kruskal_p_time < 0.01 ~ "**",
      kruskal_p_time < 0.05 ~ "*",
      TRUE ~ ""))

kw_results <- kw_results %>%
  mutate(Group = as.character(Group)) %>%
    mutate(Group = case_when(
      Group == "group0" ~ "Control group",
      Group == "group2" ~ "Involved group",
      TRUE ~ Group)) %>%
    rename(Scale = Category)


# Create the table with kable
markdown_table_kruskal <- kable(kw_results, format = "markdown",
                        col.names = c("Scale", "Group", "P-Value", "Kruskal-Wallis H"))

# Save the Markdown table to a text file
writeLines(markdown_table_kruskal, "outputs/markdown_table_kruskal.md")

# Wilcoxon rank-sum test for groups within each category and time point
wilcox_results_rq1 <- combined_df_g02 %>%
  filter(!Category %in% c("SW_Mean", "CS_Mean", "SW_CS_Mean")) %>% #exclude irrelevant categories for t
  group_by(Category, Time_Point) %>%
  summarise(
    wilcox_p_group = wilcox.test(MeanValue ~ Group)$p.value,
    W_Statistic = wilcox.test(MeanValue ~ Group)$statistic,
    .groups = 'drop'
  ) %>%
  mutate(
    significance_group = case_when(
      wilcox_p_group < 0.001 ~ "***",
      wilcox_p_group < 0.01 ~ "**",
      wilcox_p_group < 0.05 ~ "*",
      TRUE ~ ""
    )
  )

wilcox_results_rq1 <- wilcox_results_rq1 %>%
  rename(Scale = Category)


# Create the table with kable
markdown_table_wilcox_rq1 <- kable(wilcox_results_rq1, format = "markdown",
                                col.names = c("Scale", "Measurement Point", "P-Value", "W Statistic", "

# Save the Markdown table to a text file
writeLines(markdown_table_wilcox_rq1, "outputs/markdown_table_wilcox_rq1.md")

# Merge results back into the main dataframe
combined_df_g02 <- combined_df_g02 %>%
```

```r
  left_join(kw_results, by = c("Category", "Group")) %>%
  left_join(wilcox_results, by = c("Category", "Time_Point"))

# Calculate the means for each Group, Competence, and TimePoint
# adding error bars
df_means <- combined_df_g02 %>%
  filter(!Category %in% c("SW_Mean", "CS_Mean", "SW_CS_Mean")) %>% #exclude irrelevant categories for t
  group_by(Group, Category, Time_Point) %>%
  summarise(MeanValue2 = mean(MeanValue),
            LowerCI = MeanValue2 - qt(0.975, length(MeanValue) - 1) * sd(MeanValue) / sqrt(length(MeanVa
            UpperCI = MeanValue2 + qt(0.975, length(MeanValue) - 1) * sd(MeanValue) / sqrt(length(MeanVa
            .groups = 'drop') %>%
  left_join(kw_results, by = c("Category", "Group")) %>%
  left_join(wilcox_results, by = c("Category", "Time_Point"))

# Define a labeller function to rename facets
facet_labeller <- labeller(Category = c(
  AT_Mean = "Sustainability Attitudes\n(Attitude)",
  INT_Mean = "Sustainability Attitudes\n(Intention)",
  PBC_Mean = "Sustainability Attitudes\n(PBC)",
  SN_Mean = "Sustainability Attitudes\n(Subjective norms)",
  B_Mean = "Sustainability Behaviours\n(Behaviour)",
  TPB_Mean = "Cumulative\nSA and SB"))

df_means1 <- df_means %>%
  mutate(Group = recode(Group, "group0" = "Control group", "group2" = "Involved group")) %>%
  mutate(Time_Point = recode(Time_Point, "t1" = "MP1", "t2" = "MP2", "t3" = "MP3"))


# Custom colors for the groups
custom_colors <- c("#E7298A", "#1B9E77")

# Add a new factor level to control the order and size of the facets
df_means1$Category <- factor(df_means1$Category, levels = c("AT_Mean", "INT_Mean", "B_Mean", "PBC_Mean"

# Create the graph
(rq1_graph_perfect <- ggplot(df_means1, aes(x = Time_Point, y = MeanValue2, group = Group, color = Grou
  geom_line() +
  geom_point() +
  geom_errorbar(aes(ymin = LowerCI, ymax = UpperCI), width = 0.1) +  # Adding error bars
  facet_wrap(~ Category, scales = "fixed", labeller = facet_labeller, ncol = 3) +  # 3x2 grid layout
  labs(x = "\nMeasurement Points", y = "\nMean Value") +
  scale_color_manual(values = custom_colors) +  # Custom colors for the groups
  theme_minimal() +
  theme(panel.grid.major = element_blank(),   # Remove major grid lines
        panel.grid.minor = element_blank(),   # Remove minor grid lines
        panel.border = element_rect(fill = NA, color = "black"), # Add border around each facet
        legend.position = "bottom",
        legend.text = element_text(size = 12, face = "italic"),
        legend.title = element_text(size = 12, face = "bold"),
        axis.title.x = element_text(size = 14, face = "plain"),
        axis.title.y = element_text(size = 14, face = "plain")) +
  geom_text_repel(data = df_means_grouped %>% filter(significance_group != ""),
```

```
                  aes(x = Time_Point, y = MeanValue2, label = significance_group),
                  vjust = -1.5, color = "black", size = 6, inherit.aes = FALSE, segment.color = NA) +
  geom_text_repel(data = df_means_grouped %>% filter(significance_time != ""),
                  aes(x = Time_Point, y = MeanValue2, label = significance_time),
                  vjust = -1.0, color = "blue", size = 3, inherit.aes = FALSE) +
  guides(color = guide_legend(title = NULL)))  # Remove the legend title

ggsave(rq1_graph_perfect, file = "outputs/rq1_graph_perfect.png", width = 7, height = 5)


### RQ2 relationship Efficacy and SA/SB (TPB) with spearman correlation at MZP3 ----
# Shapiro-Wilk normality test
shapiro.test(mzp3_cleaner$TPB_mean) #normally distributed p>0.05
shapiro.test(mzp3_cleaner$SW_mean) #not normally distributed p<0.05
shapiro.test(mzp3_cleaner_cs$CS_mean) #not normally distributed p<0.05


# Spearman correlation
spearman_test <- cor.test(mzp3_cleaner$TPB_mean, mzp3_cleaner$SW_mean, method = "spearman")
print(spearman_test)
spearman_cor <- spearman_test$estimate
spearman_pval <- spearman_test$p.value
# p value rejects 0 hypothesis of no correlation -> relevant
# spearman_correlation 0.794 -> strong relationship

# Plot with annotation (SPEARMAN)
(rq2a_graph_cor_tpb_sw <-ggplot(mzp3_cleaner, aes(x = TPB_mean, y = SW_mean)) +
  geom_point(alpha = 0.5, size = 2, color = "#7CFC00") +
  geom_smooth(method = "lm", se = FALSE, color = "#A6761D") +
  ylim(0, 3) +
  theme_clean() +
    annotate("text", size = 3, x = 1, y = 2.5, label = paste("Spearman's rho:", round(spearman_cor, 2),
  labs(x = "\nMean Sustainability Attitudes and Behaviours (TPB)", y = "Mean Efficacy beliefs\n"))

ggsave(rq2a_graph_cor_tpb_sw, file = "outputs/rq2a_graph_cor_tpb_sw.png", width = 7, height = 5)

### RQ3a: Comparing personal and collective efficacy between and within groups at MP3 ----
# data preparation
# Read the Excel file for classifications
codebook_sw_cs <- read_excel("data/data_collection/codebook_sw_cs.xlsx")

# Specify the questions of interest explicitly
questions_sw <- c("CS01_01", "CS01_02", "CS01_03", "SW01_01", "SW01_02", "SW01_03_inverse", "SW01_04",

# Calculate the mean scores for the specified questions
mean_scores_sw <- colMeans(mzp3_cleaner[, questions_sw], na.rm = TRUE)

# Convert the mean scores to a data frame for plotting
mean_scores_sw_df <- data.frame(
  Question = names(mean_scores_sw),
  Mean_Score = mean_scores_sw
)
```

```r
# Join the dataframes by the column 'Question'
merged_sw_cs <- left_join(mean_scores_sw_df, codebook_sw_cs, by = "Question")


### comparing between groups SW/CS ----
# check distribution of data
# Perform Shapiro-Wilk tests
shapiro_results_sw_cs <- data.frame(
  Question = questions_sw,
  p_value = sapply(questions_sw, function(question) shapiro.test(mzp3_cleaner[[question]])$p.value)
)
## none is normally distributed

# filter out relevant categories
df_tp3 <- combined_df_g02 %>%
  filter(Time_Point == "t3" & Category %in% c("SW_Mean", "CS_Mean"))


### checking wilcoxon for differences in personal/collective between groups
wilcox_rq3_sw_cs_between <- df_tp3 %>%
  group_by(Category) %>%
  rename(Scale = Category) %>%
  summarise(wilcox_p_group = wilcox.test(MeanValue ~ Group)$p.value,
            W_Statistic = wilcox.test(MeanValue ~ Group)$statistic,
    .groups = 'drop') #%>%
  mutate(significance_group = case_when(
      wilcox_p_group < 0.001 ~ "***",
      wilcox_p_group < 0.01 ~ "**",
      wilcox_p_group < 0.05 ~ "*",
      TRUE ~ ""))
# no significant differences between the groups

# Create the table with kable
markdown_table_wilcox_rq3_sw_cs_between <- kable(wilcox_rq3_sw_cs_between, format = "markdown",
                                    col.names = c("Scale", "P-Value", "W Statistic"))

# Save the Markdown table to a text file
writeLines(markdown_table_wilcox_rq3_sw_cs_between, "outputs/markdown_table_wilcox_rq3_sw_cs_between.md

# Calculate means for each Group and Competence
df_means3 <- df_tp3 %>%
  filter(!is.na(MeanValue)) %>%
  group_by(Group, Category) %>%
  summarise(
    MeanValue2 = mean(MeanValue),
    LowerCI = MeanValue2 - qt(0.975, length(MeanValue) - 1) * sd(MeanValue) / sqrt(length(MeanValue)),
    UpperCI = MeanValue2 + qt(0.975, length(MeanValue) - 1) * sd(MeanValue) / sqrt(length(MeanValue)),
    .groups = 'drop') %>%
  left_join(wilcox_results_sw_cs, by = c("Category"))


# comparing personal/collective within groups ----
# Aggregate scores for CS and SW questions within each group
```

```r
aggregate_scores <- mzp3_cleaner %>%
  pivot_longer(cols = c(starts_with("CS"), starts_with("SW")), names_to = "Question", values_to = "Score
  mutate(Type = ifelse(str_detect(Question, "CS"), "CS", "SW")) %>%
  drop_na() %>%
  group_by(Group, Type) %>%
  summarize(Aggregate_Score = mean(Score, na.rm = TRUE), .groups = 'drop')

# Reshape data to wide format to ensure paired observations
wide_data <- aggregate_scores %>%
  pivot_wider(names_from = Type, values_from = Aggregate_Score)

# Function to perform Wilcoxon signed-rank test for aggregated scores
wilcoxon_test_aggregated <- function(data, group) {
  filtered_data <- data %>% filter(Group == group)

  cs_scores <- filtered_data$CS
  sw_scores <- filtered_data$SW

  test <- wilcox.test(cs_scores, sw_scores, paired = TRUE)
  return(data.frame(
    Group = group,
    Statistic = test$statistic,
    P_Value = test$p.value
  ))
}

# Perform Wilcoxon tests for overall comparison within each group
results_group0_overall <- wilcoxon_test_aggregated(wide_data, "group0")
results_group2_overall <- wilcoxon_test_aggregated(wide_data, "group2")

# Combine results into a dataframe
wilcoxon_results_overall <- rbind(results_group0_overall, results_group2_overall)

# Add Significance column for stars
wilcoxon_results_overall$Significance <- ifelse(wilcoxon_results_overall$P_Value < 0.05, "*", "")

# create table
wilcoxon_rq3_sw_cs_within <- wilcoxon_results_overall %>%
  mutate(Group = as.character(Group)) %>%
  mutate(Group = case_when(
    Group == "group0" ~ "Control group",
    Group == "group2" ~ "Involved group",
    TRUE ~ Group))

# Create the table with kable
markdown_table_wilcoxon_rq3_sw_cs_within <- kable(wilcoxon_rq3_sw_cs_within, format = "markdown",
                                                  col.names = c("Group", "W Statistic", "P-Value"))

# Save the Markdown table to a text file
writeLines(markdown_table_wilcoxon_rq3_sw_cs_within, "outputs/markdown_table_wilcox_rq3_sw_cs_within.md

# Prepare data for visualization
# Function to calculate confidence intervals
```

```r
calculate_ci <- function(data, conf_level = 0.95) {
  mean_val <- mean(data, na.rm = TRUE)
  stderr <- sd(data, na.rm = TRUE) / sqrt(length(data))
  error_margin <- qnorm((1 + conf_level) / 2) * stderr
  lower_ci <- mean_val - error_margin
  upper_ci <- mean_val + error_margin
  return(data.frame(Mean = mean_val, Lower_CI = lower_ci, Upper_CI = upper_ci))
}

# Apply CI calculation to the mean scores
mean_scores_with_ci <- mzp3_cleaner %>%
  pivot_longer(cols = c(starts_with("CS"), starts_with("SW")), names_to = "Question", values_to = "Score
  mutate(Type = ifelse(str_detect(Question, "CS"), "CS", "SW")) %>%
  drop_na() %>%
  group_by(Group, Type) %>%
  summarize(Mean_Score = mean(Score, na.rm = TRUE),
            Lower_CI = mean(Score, na.rm = TRUE) - qnorm(0.975) * (sd(Score, na.rm = TRUE) / sqrt(n())),
            Upper_CI = mean(Score, na.rm = TRUE) + qnorm(0.975) * (sd(Score, na.rm = TRUE) / sqrt(n())),


mean_scores_with_ci1 <- mean_scores_with_ci %>%
  mutate(Group = recode(Group, "group0" = "Control group", "group2" = "Involved group")) %>%
  mutate(Type = recode(Type, "CS" = "Collective efficacy beliefs", "SW" = "Personal efficacy beliefs"))

# Define custom colors2
custom_colors2 <- c("#D95F02", "#E6AB02")

# Plotting the results with annotation and error bars
(rq2b_boxplot_sw_cs_overall <- ggplot(mean_scores_with_ci1, aes(x = Type, y = Mean_Score, fill = Type))
  geom_bar(stat = "identity", position = position_dodge(), width = 0.7) +
  geom_errorbar(aes(ymin = Lower_CI, ymax = Upper_CI), width = 0.2, position = position_dodge(0.7)) +
  facet_wrap(~ Group, scales = "fixed") +
  labs(y = "\n\nMean Value") +
  theme_clean() +
  scale_fill_manual(values = custom_colors2) +
  theme(axis.title.x = element_blank(),
          axis.text.x = element_blank(),
          axis.ticks.x = element_blank(),
          legend.position = "bottom")+
    guides(fill = guide_legend(title = NULL)))

ggsave(rq2b_boxplot_sw_cs_overall, file = "outputs/rq2b_boxplot_sw_cs_overall.png", width = 7, height =


### RQ3b: Comparing aim- and action-focused efficacy between and within groups at MP3 ----
# comparing between groups aim vs action focussed ----
# add new classification only based on aim vs action
merged_group_scores <- merged_group_scores %>%
  mutate(Theoretical_classification2 = case_when(
    grepl("aim", Theoretical_classification, ignore.case = TRUE) ~ "aim",
    TRUE ~ "action"
  ))
```

```r
# compare within theoretical classifications:
final_analysis2 <- merged_group_scores %>%
  group_by(Theoretical_classification2, Group) %>%
  summarize(Avg_Mean_Score2 = mean(Mean_Score, na.rm = TRUE), .groups = 'drop')


# only two groups to compare within each theoretical classification
wilcoxon_results_aim_action <- merged_group_scores %>%
  group_by(Theoretical_classification2) %>%
  summarize(
    W_test = list(wilcox.test(Mean_Score ~ Group, data = cur_data())),
    .groups = 'drop'
  )

# Extract p-values and test statistics
wilcoxon_results_aim_action$P_Value <- sapply(wilcoxon_results_aim_action$W_test, function(x) x$p.value)
wilcoxon_results_aim_action$Statistic <- sapply(wilcoxon_results_aim_action$W_test, function(x) x$statis

# Add significance stars
wilcoxon_results_aim_action$Significance <- ifelse(wilcoxon_results_aim_action$P_Value < 0.05, "*", "")

# create table
wilcox_rq3_aim_action_between <- wilcoxon_results_aim_action %>%
  select(-W_test)

markdown_table_wilcox_rq3_aim_action_between <- kable(wilcox_rq3_aim_action_between, format = "markdown
                                                      col.names = c("Theoretical classification", "P-Value",


# Save the Markdown table to a text file
writeLines(markdown_table_wilcox_rq3_aim_action_between, "outputs/markdown_table_wilcox_rq3_aim_action_

# Merge Wilcoxon test results back into the main dataframe for plotting
merged_group_scores <- merged_group_scores %>%
  left_join(wilcoxon_results_aim_action %>% select(Theoretical_classification2, Significance), by = "The

# Apply CI calculation to the mean scores
mean_scores_with_ci2 <- merged_group_scores %>%
  group_by(Group, Theoretical_classification2) %>%
  drop_na() %>%
  summarize(
    Mean_Score = mean(Score, na.rm = TRUE),
    Lower_CI = mean(Score, na.rm = TRUE) - qnorm(0.975) * (sd(Score, na.rm = TRUE) / sqrt(n())),
    Upper_CI = mean(Score, na.rm = TRUE) + qnorm(0.975) * (sd(Score, na.rm = TRUE) / sqrt(n())),
    .groups = 'drop')

mean_scores_with_ci2 <- mean_scores_with_ci2 %>%
  mutate(Group = recode(Group, "group0" = "Control group", "group2" = "Involved group")) %>%
  mutate(Theoretical_classification2 = recode(Theoretical_classification2, "action" = "Action-focused e
#rename(`Types of Efficacy` = Type)

# Define custom colors2
custom_colors3 <- c("#7570B3", "#666666")
```

```r
# Making pretty graph
(rq2b_boxplot_aim_action <- ggplot(mean_scores_with_ci2, aes(x = Theoretical_classification2, y = Mean_S
    geom_bar(stat = "identity", position = position_dodge(), width = 0.7) +
    geom_errorbar(aes(ymin = Lower_CI, ymax = Upper_CI), width = 0.2, position = position_dodge(0.7)) +
    facet_wrap(~ Group, scales = "fixed") +
    labs(y = "\n\nMean Value") +
    theme_clean() +
    scale_fill_manual(values = custom_colors3) +
    theme(axis.title.x = element_blank(),
          axis.text.x = element_blank(),
          axis.ticks.x = element_blank(),
          legend.position = "bottom")+
    guides(fill = guide_legend(title = NULL)))

ggsave(rq2b_boxplot_aim_action, file = "outputs/rq2b_boxplot_aim_action.png", width = 7, height = 5)

### compare action vs aim within groups ----
# Aggregate scores for aim and action questions within each group
aggregate_scores <- merged_group_scores %>%
  group_by(Group, Theoretical_classification2) %>%
  summarize(Aggregate_Score = mean(Mean_Score, na.rm = TRUE), .groups = 'drop')

# Reshape data to wide format to ensure paired observations
wide_data2 <- aggregate_scores %>%
  pivot_wider(names_from = Theoretical_classification2, values_from = Aggregate_Score)

# Function to perform Wilcoxon signed-rank test for aggregated scores
wilcoxon_test_aggregated2 <- function(data, group) {
  filtered_data <- data %>% filter(Group == !!group)

  aim_scores <- filtered_data$aim
  action_scores <- filtered_data$action

  test <- wilcox.test(aim_scores, action_scores, paired = TRUE)
  return(data.frame(
    Group = group,
    Statistic = test$statistic,
    P_Value = test$p.value
  ))
}

# Perform Wilcoxon tests for overall comparison within each group
results_group0_aa <- wilcoxon_test_aggregated2(wide_data2, "group0")
results_group2_aa <- wilcoxon_test_aggregated2(wide_data2, "group2")

# Combine results into a dataframe
wilcoxon_results_aim_action2 <- rbind(results_group0_aa, results_group2_aa)

# Add Significance column for stars
wilcoxon_results_aim_action2$Significance <- ifelse(wilcoxon_results_aim_action2$P_Value < 0.05, "*", "

# create table
wilcoxon_rq3_aim_action_within <- wilcoxon_results_aim_action2 %>%
```

```r
  mutate(Group = as.character(Group)) %>%
  mutate(Group = case_when(
    Group == "group0" ~ "Control group",
    Group == "group2" ~ "Involved group",
    TRUE ~ Group))

# Create the table with kable
markdown_table_wilcoxon_rq3_aim_action_within <- kable(wilcoxon_rq3_aim_action_within, format = "markdo
                                                       col.names = c("Group", "W Statistic", "P-Value"))

# Save the Markdown table to a text file
writeLines(markdown_table_wilcoxon_rq3_aim_action_within, "outputs/markdown_table_wilcoxon_rq3_aim_acti
```