```
---
title: "MPxMA_Replication"
output: html_document
date: "2024-11-05"
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

```{r libraries, include = FALSE}
#install.packages(c("readxl", "tidyverse", "formattable", "lme4", "lmerTest",
#                   "writexl",  "irr", "sjPlot", "sjstats", "apaTables",
#                   "Hmisc", "dplyr"), dependencies = TRUE)
#install.packages('dunn.test', repos="http://cran.us.r-project.org")

require(dunn.test)
require(readxl)
require(tidyverse)
require(formattable)
require(lme4)
require(lmerTest) #for p-values
require(writexl) # for exporting final excel
require(irr)
require(sjstats) # for tau 11
require(dplyr)
require(apaTables)
require(ggplot2)
require(plotly)
library(factoextra)
library(DemographicTable)
library(gtsummary)
library(cluster)  # clustering algorithms and Silhouette Score
library(car)


rm(list = ls())
setwd("~/Documents/Data_Analysis/MPxMA_Replication_poster")

```

```{r vis settings, echo = FALSE}

# Define the Wes Anderson colors
#install.packages('wesanderson', repos="http://cran.us.r-project.org")
library(wesanderson)
wes_colors <- wes_palette(n = 5, name = "AsteroidCity1")

```

```{r Loading all data, include = FALSE}

### Loading data

data <- read_excel("EHR_student_level.xlsx")

# Choosing columns
data <- data [, (colnames(data) %in%
  c('PRE_SC', 'PRE_TOTAL_RT', 'PRE_AVG_RT',
    'MA_ITM_PCT', 'MA_TOTAL_SC', 'MA_NR_SC', 'MA_NC_SC', 'MA_WO_SC',
    'MV_ITM_PCT', 'MV_TOTAL_SC', 'MV_AT_SC', 'MV_IN_SC', 'MV_UT_SC',
    'StuID', 'Gender_Female', 'Gifted', 'ELL', 'Race_Ethnicity'))]

### --- Delete cases with NAs in Pre Math anxiety or Pre Math performance
```
```

```
data <- data %>% drop_na(MA_TOTAL_SC)
data <- data %>% drop_na(PRE_SC)

```

Sample desctiptives

Correlation matrix

```{r cor matrix, echo=FALSE}

## Quantitative variables
Replication_quant_with_action <-
  data [, (colnames(data) %in%
             c('PRE_SC', 'MA_TOTAL_SC'))]

names(Replication_quant_with_action)[names(Replication_quant_with_action) == 'PRE_SC'] <-
'Math_perform'
names(Replication_quant_with_action)[names(Replication_quant_with_action) ==
'MA_TOTAL_SC'] <- 'Math_anxiety'

# Creating matrix
apa.cor.table(
  data = Replication_quant_with_action,
  filename = "Descriptives.doc",
  table.number = 1,
  show.conf.interval = TRUE,
  show.sig.stars = TRUE,
  landscape = TRUE
)

```

Demographics

```{r sample demographics, echo=FALSE}

data$Gifted <- as.factor(data$Gifted)
data$ELL <- as.factor(data$ELL)

# Table with all stats (does not knitted in RMarkdown)
descriptives <- DemographicTable(data=data, include = c('Gender_Female', 'Gifted', 'ELL',
'Race_Ethnicity'))

# Table for RMarkdown
data %>%
  dplyr::select(c('Gender_Female', 'Gifted', 'ELL', 'PRE_SC','MA_TOTAL_SC',
'Race_Ethnicity')) %>%
  tbl_summary()

```

Choosing number of clusters

Elbow method

```{r RQ1, echo=TRUE}

# Z-scoring MP and MA
data$PRE_SC_z <-
  (data$PRE_SC - mean(data$PRE_SC))/sd(data$PRE_SC)
data$MA_TOTAL_SC_z <-
  (data$MA_TOTAL_SC - mean(data$MA_TOTAL_SC))/sd(data$MA_TOTAL_SC)

# Creating new dataframes for PRE-levels clustering based on scaled variables
```

```
PRE_z <- data %>% as.data.frame() %>%
  dplyr::select(PRE_SC_z, MA_TOTAL_SC_z)

### --- How many clusters - Elbow method (widely used, recommended)
fviz_nbclust(PRE_z, kmeans, method = "wss") +
  geom_vline(xintercept = 4, linetype = 2)+
  labs(subtitle = "Elbow method")
```

Silhouette scores

```{r RQ1 choosing clusters with Silhouette score, echo=TRUE}
# Range of cluster numbers to test
max_clusters <- 10
silhouette_scores <- numeric(max_clusters)

# Loop through different numbers of clusters
for (k in 2:max_clusters) {
  set.seed(123)  # For reproducibility
  kmeans_result <- kmeans(PRE_z, centers = k)
  sil <- silhouette(kmeans_result$cluster, dist(PRE_z))
  silhouette_scores[k] <- mean(sil[, 3])  # Average Silhouette score for this k
}

# Find the number of clusters with the highest average Silhouette score
best_k <- which.max(silhouette_scores)
cat("The optimal number of clusters is", best_k, "with an average Silhouette score of",
silhouette_scores[best_k], "\n")

# Plot the Silhouette scores for each number of clusters
plot(2:max_clusters, silhouette_scores[2:max_clusters], type = "b",
     xlab = "Number of Clusters", ylab = "Average Silhouette Score",
     main = "Silhouette Score for Different Numbers of Clusters")
```

Clustering with 4 centers

```{r RQ1 clustering, include = TRUE}

### --- Applying k-means clustering
set.seed(20)
cluster <- kmeans(PRE_z, centers = 4, nstart = 25) # put the optimal number of clusters in
"centers"
print(cluster)

# Save the cluster number in the dataset as column 'cluster_results'
data$cluster_results <- as.factor(cluster$cluster)
```

```{r RQ1 name clusters, include = FALSE}

## Saving clusters mean MP and MA values
data <- data %>%
 group_by(cluster_results) %>%
  mutate(PreMP_mean = mean(PRE_SC),
            MA_mean = mean(MA_TOTAL_SC)) %>%
  ungroup()

## Saving clusters names based on mean MP and MA values
# Put in MP levels
data$pre_MP_group <-
  ifelse(data$PreMP_mean<mean(data$PRE_SC, na.rm=TRUE),
```

```
        "lMP", "hMP")
# Put in MA levels
data$MA_group <-
  ifelse(data$MA_mean<mean(data$MA_TOTAL_SC, na.rm=TRUE),
         "lMA", "hMA")
# Combining MP and MA levels into one var
data$cluster_groups <-
  paste(data$pre_MP_group, data$MA_group, sep="_")

## Saving clusters as factors with appropriate levels
data$cluster_groups <-
  factor(data$cluster_groups,
         levels = c("lMP_hMA", "lMP_lMA", "hMP_lMA", "hMP_hMA"))

## Calculating means in clusters to check if they are correct
data %>%
 group_by(cluster_groups) %>%
  summarise(PreMP_mean = mean(PRE_SC),
            PreMP_sd = sd(PRE_SC),
            MA_mean = mean(MA_TOTAL_SC),
            MA_sd = sd(MA_TOTAL_SC))

# To compare to the best group
data$cluster_groups_best <-
  factor(data$cluster_groups,
         levels = c("hMP_lMA", "hMP_hMA", "lMP_lMA", "lMP_hMA"))

```

Visualizing clusters

```{r RQ1 vis with centroids, echo=TRUE}
# Calculate centroids from your K-means result
centroids <- as.data.frame(cluster$centers)

cluster_colors <- c("#E69F00", "#56B4E9", "#009E73", "#F0E442")

# Visualize the data with ggplot
library(ggplot2)
ggplot(data, aes(MA_TOTAL_SC_z, PRE_SC_z)) +
  geom_jitter(aes(color = factor(cluster_groups))) +
  geom_point(data = centroids, aes(x = MA_TOTAL_SC_z, y = PRE_SC_z),
             color = "black", size = 4, shape = 8) +  # Red stars for centroids
  scale_color_manual(values = cluster_colors) +
  labs(color = "Cluster", x = "Math Anxiety Score", y = "Math Test Score") +
  theme_minimal()

```


Clusters' demographics

```{r clusters demographics, echo=FALSE}

data$Gifted <- as.factor(data$Gifted)
data$ELL <- as.factor(data$ELL)

# Table with all stats (does not knitted in RMarkdown)
descriptives <- DemographicTable(data=data, groups = 'cluster_groups', include =
c('Gender_Female', 'Gifted', 'ELL'))

# Table for RMarkdown
data %>%
  dplyr::select(c('Gender_Female', 'Gifted', 'ELL', 'cluster_groups',
'PRE_SC','MA_TOTAL_SC')) %>%
```

```
  tbl_summary(by='cluster_groups')

# Table with PreMP and PreMA means and sds
data %>%
    group_by(cluster_groups) %>%
    summarise_at( c('PRE_SC','MA_TOTAL_SC'), c(mean = mean, sd = sd))

```

Comparison by MP

```{r comparison of MP, echo=FALSE}

# Checking normality
data %>%
  group_by(cluster_groups) %>%
  summarise(shapiro_statistic = shapiro.test(PRE_SC)$statistic,
            p.value = shapiro.test(PRE_SC)$p.value)

# Checking homogeneity of variance
leveneTest(PRE_SC ~ cluster_groups, data = data)
bartlett.test(PRE_SC ~ cluster_groups, data = data)

## MP comparison
dunn.test(data$PRE_SC, g=data$cluster_groups, method='bonferroni')
```

Comparison by MA

```{r comparison of MA, echo=FALSE}

# Checking normality
data %>%
  group_by(cluster_groups) %>%
  summarise(shapiro_statistic = shapiro.test(MA_TOTAL_SC)$statistic,
            p.value = shapiro.test(MA_TOTAL_SC)$p.value)

# Checking homogeneity of variance
leveneTest(MA_TOTAL_SC ~ cluster_groups, data = data)
bartlett.test(MA_TOTAL_SC ~ cluster_groups, data = data)

## MA comparison
dunn.test(data$MA_TOTAL_SC, g=data$cluster_groups, method='bonferroni')

```

Vizualization of comparison by MP and MA (z-scored)

```{r vis comparison of MP and MA, echo=FALSE}

## Visualization for both

# Creating long format table
data_long <- pivot_longer(data,
                              cols = c('PRE_SC_z', 'MA_TOTAL_SC_z'),
                              names_to = 'Variable',
                              values_to = 'Value')

# Specify levels for factor "Variable" (so MP goes first on the viz)
data_long$Variable <- factor(data_long$Variable , levels=c("PRE_SC_z", "MA_TOTAL_SC_z"))


# Create a boxplot for each variable with facets for clusters
ggplot(data_long, aes(x = Variable , y = Value, fill = Variable)) +
  geom_boxplot() +
```

```
  labs(x = "Cluster", y = "Value") +
  facet_wrap(~ cluster_groups_best, scales = "fixed") +
  scale_fill_manual(values = wes_colors) +
  theme_minimal()
```

MP distribution

```{r MP distribution, echo=FALSE}

# Create the ggplot2 density plot
p_PRE_SC <- ggplot(data,
                aes(x = PRE_SC, fill = cluster_groups)) +
        geom_density(alpha = 0.8) +
        scale_fill_manual(values = cluster_colors) +
        theme_minimal()

# Convert the ggplot object to a plotly object
p_plotly_PRE_SC <- ggplotly(p_PRE_SC, tooltip = "fill")

# Make the plotly plot interactive such that hovering over the legend highlights the
specific category
p_plotly_PRE_SC %>%
  style(hoverinfo = "none", hoveron = "points", traces = c(1,2)) %>%
  layout(showlegend = TRUE)


```


MA distribution

```{r MA distribution, echo=FALSE}

# Create the ggplot2 density plot
p_MA_TOTAL_SC <- ggplot(data,
                aes(x = MA_TOTAL_SC, fill = cluster_groups)) +
        geom_density(alpha = 0.8) +
        scale_fill_manual(values = cluster_colors) +
        theme_minimal()

# Convert the ggplot object to a plotly object
p_plotly_MA_TOTAL_SC <- ggplotly(p_MA_TOTAL_SC, tooltip = "fill")

# Make the plotly plot interactive such that hovering over the legend highlights the
specific category
p_plotly_MA_TOTAL_SC %>%
  style(hoverinfo = "none", hoveron = "points", traces = c(1,2)) %>%
  layout(showlegend = TRUE)

```