MPxMA profiles initial

Alena

8/29/2023

Sample desctiptives

Correlation matrix

```
##
##
## Table 1
##
## Means, standard deviations, and correlations with confidence intervals
##
##
##   Variable  M     SD   1            2
##   1. PreMP  4.60  2.65
##
##   2. PreMA  13.58 6.00 -.25**
##                        [-.30, -.20]
##
##   3. PreMSE 3.50  0.96 .36**        -.58**
##                        [.31, .41]   [-.61, -.54]
##
##
## Note. M and SD are used to represent mean and standard deviation, respectively.
## Values in square brackets indicate the 95% confidence interval.
## The confidence interval is a plausible range of population correlations
## that could have caused the sample correlation (Cumming, 2014).
##  * indicates p < .05. ** indicates p < .01.
##
```

Sample demographics

Characteristic	**N = 1,649**[1]
Gender	
Female	782 (47%)
Male	867 (53%)
Unknown	0 (0%)
race_ethnicity	
American Indian/Alaska Native	10 (0.6%)

Asian	406 (25%)
Black/African American	82 (5.0%)
Hispanic/Latino	268 (17%)
Native Hawaiian or Other Pacific Islander	0 (0%)
Two or more races	52 (3.2%)
White	806 (50%)
Unknown	25
IEP	
0	1,461 (89%)
1	188 (11%)
EIP	
0	1,535 (93%)
1	114 (6.9%)
GIFTED	
0	1,387 (84%)
1	262 (16%)
ESOL	
0	1,485 (90%)
1	164 (9.9%)
PreMP	4 (3, 7)
Unknown	357
PreMA	14 (9, 18)
Unknown	443
PreMSE	3.60 (3.00, 4.20)
Unknown	448

[1] n (%); Median (Q1, Q3)

Analysis

Clustering

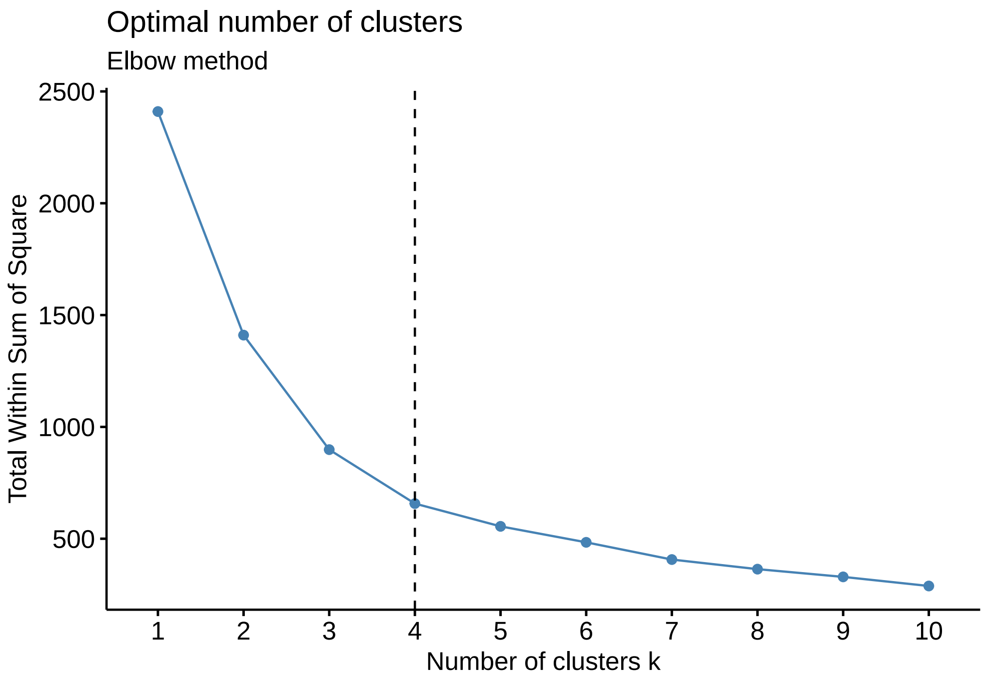
Choosing number of clusters

```
### --- Delete cases with NAs in Pre Math anxiety or Pre Math performance
FH2T <- FH2T %>% drop_na(PreMA)
FH2T <- FH2T %>% drop_na(PreMP)

# Z-scoring MP and MA
FH2T$PreMP_z <-
  (FH2T$PreMP - mean(FH2T$PreMP))/sd(FH2T$PreMP)
FH2T$PreMA_z <-
  (FH2T$PreMA - mean(FH2T$PreMA))/sd(FH2T$PreMA)

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

```
PRE_z <- FH2T %>% as.data.frame() %>%
  dplyr::select(PreMA_z, PreMP_z)
```

Dropping NAs and Z-scoring

```
fviz_nbclust(PRE_z, kmeans, method = "wss") +
  geom_vline(xintercept = 4, linetype = 2)+
  labs(subtitle = "Elbow method")
```



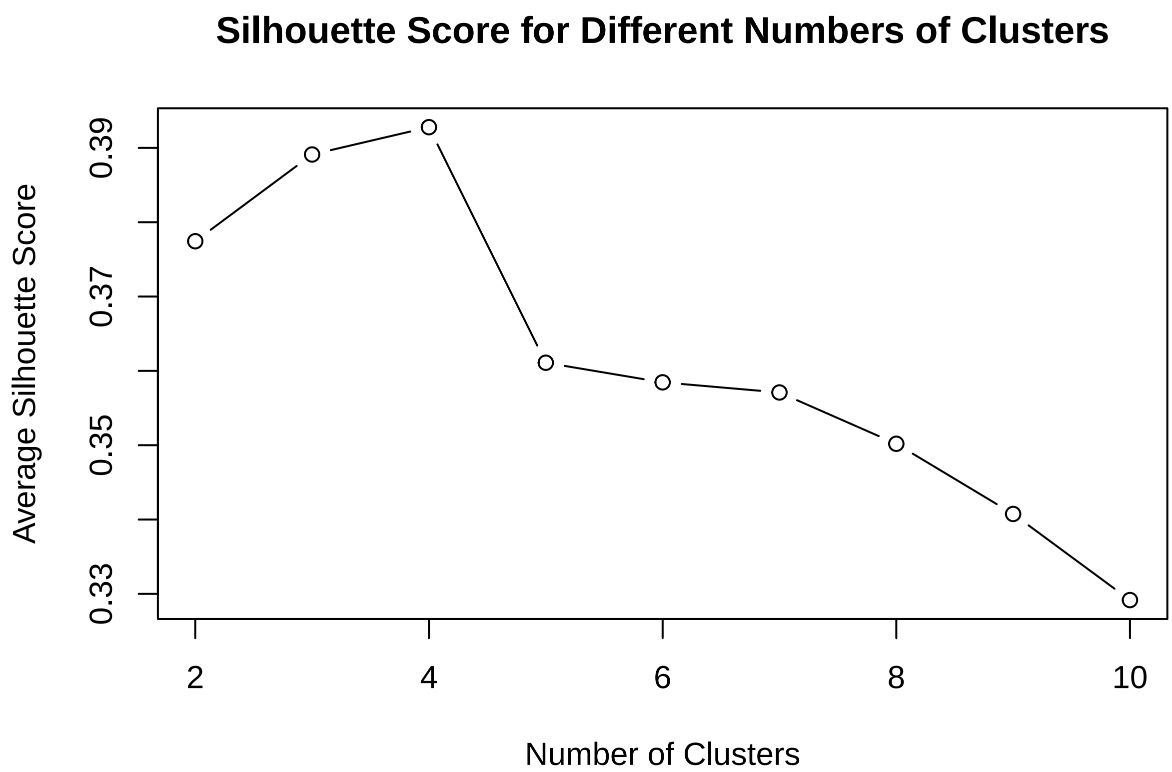
Elbow method

```
# Range of cluster numbers to test
silhouette_scores <- numeric(10)

# Loop through different numbers of clusters
for (k in 2:10) {
  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_scor
```

Silhouette scores

```
## The optimal number of clusters is 4 with an average Silhouette score of 0.3927846
```

```r
# Plot the Silhouette scores for each number of clusters
plot(2:10, silhouette_scores[2:10], type = "b",
     xlab = "Number of Clusters", ylab = "Average Silhouette Score",
     main = "Silhouette Score for Different Numbers of Clusters")
```



Clustering with 4 centers

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

```
## K-means clustering with 4 clusters of sizes 227, 261, 375, 343
##
## Cluster means:
##      PreMA_z    PreMP_z
## 1 -1.0654934  1.3278878
## 2  0.5822366  0.7544672
## 3 -0.6699864 -0.6206387
## 4  0.9946010 -0.7743643
##
```

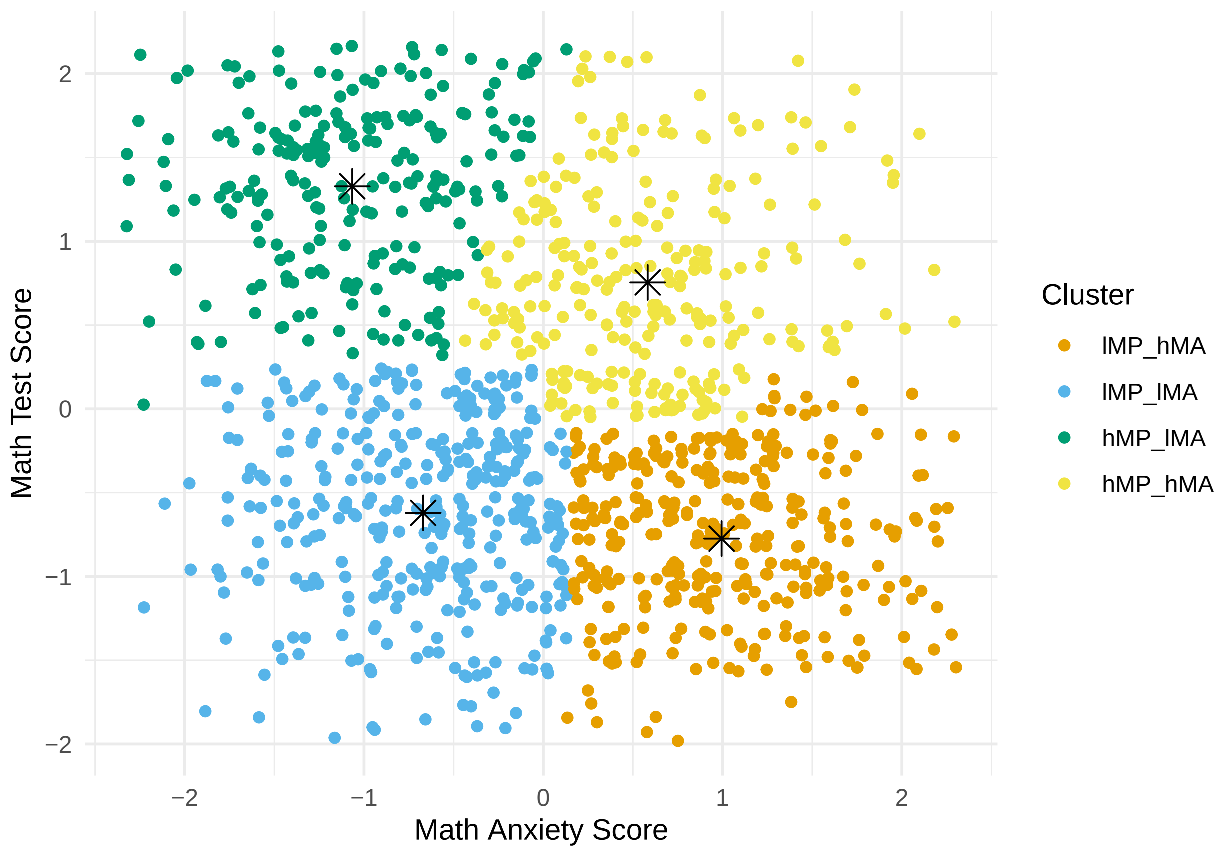
```
## Clustering vector:
##    [1] 1 1 4 2 3 4 3 3 3 1 3 1 3 3 3 4 4 4 4 4 3 2 3 4 3 2 3 3 3 3 4 3 3 4 3 3 3
##   [38] 3 3 3 3 3 3 4 2 3 4 4 4 4 4 3 4 1 2 2 2 4 3 2 2 2 4 4 3 4 4 2 2 4 3 3 4 4
##   [75] 2 2 2 2 2 3 4 3 4 3 3 4 4 3 3 3 3 4 4 4 4 3 3 3 2 4 4 2 4 4 4 3 4 4 4 4 4
##  [112] 2 4 4 4 2 4 3 4 3 4 4 2 4 3 2 4 4 4 4 4 4 4 4 2 4 4 2 4 4 3 2 4 4 2 2 3 2
##  [149] 2 1 3 3 4 4 3 4 3 2 4 3 3 4 2 4 4 4 1 3 3 4 2 4 3 3 2 4 1 3 3 4 4 3 4 3 3
##  [186] 4 3 3 2 3 3 4 4 3 4 4 1 2 4 4 3 4 3 3 4 2 2 2 2 2 1 4 1 1 4 1 3 2 4 1 1 2
##  [223] 2 3 2 1 3 2 1 3 3 3 4 3 3 4 2 2 1 2 1 4 2 2 2 1 3 3 3 1 2 4 4 4 4 4 3 3 4
##  [260] 4 3 4 4 4 3 3 3 3 4 3 4 4 3 4 4 3 3 3 3 3 3 4 3 3 3 4 3 4 3 3 4 4 4 3 4 3
##  [297] 3 3 4 4 4 3 4 4 2 3 3 4 4 1 3 4 4 4 4 4 3 3 3 4 4 4 4 4 4 3 3 4 3 3 3 3 3
##  [334] 3 4 3 3 2 3 3 4 3 3 4 4 4 3 4 2 3 4 4 4 3 4 2 3 3 3 2 4 4 3 3 3 3 3 4 4 3
##  [371] 3 3 4 3 3 3 3 3 4 3 3 4 3 4 2 4 4 3 4 3 4 4 4 4 4 3 3 3 3 3 3 4 4 3 4 3 4
##  [408] 4 3 4 4 3 4 4 4 4 3 4 4 1 4 3 4 4 1 4 4 1 3 2 1 2 4 3 3 2 3 4 3 2 2 2 2 2
##  [445] 3 2 3 3 1 3 2 4 2 2 2 2 3 1 2 3 4 2 2 3 3 3 2 4 3 4 3 4 4 4 3 4 4 3 4 4 3
##  [482] 3 4 3 3 3 3 2 4 4 4 2 4 3 1 1 2 3 3 3 1 1 3 4 3 4 2 3 4 3 1 4 3 3 4 3 3 4
##  [519] 2 3 4 1 3 4 4 2 3 3 3 3 3 2 1 3 4 4 3 3 3 4 2 1 3 3 3 3 4 4 4 2 1 2 1 2 3
##  [556] 4 1 2 1 1 3 3 3 4 4 3 4 4 4 3 3 2 3 1 3 3 2 3 3 2 4 4 4 4 3 3 4 3 3 3 3 2
##  [593] 3 2 1 1 3 2 2 2 2 3 2 3 1 3 4 3 3 3 2 4 2 3 2 1 1 4 2 4 1 2 4 4 3 2 3 4 1
##  [630] 4 3 1 4 4 4 4 2 2 4 2 1 4 1 4 2 3 2 1 4 2 4 3 4 4 2 3 3 4 4 4 1 4 4 4 4 3
##  [667] 3 1 1 3 3 3 1 4 3 4 1 3 3 1 4 2 3 2 2 1 4 1 4 1 3 3 3 3 3 4 3 4 3 2 4 3 3
##  [704] 4 2 3 1 2 1 2 2 1 2 1 1 1 1 2 2 2 1 3 4 2 3 3 2 3 4 4 3 3 3 3 3 4 4 4 1 4
##  [741] 3 4 2 3 4 3 3 4 3 4 2 4 4 3 3 3 4 3 1 4 3 3 3 4 3 3 3 1 3 2 3 4 4 4 3 4 4
##  [778] 4 2 3 3 4 4 1 2 4 3 3 3 4 4 4 3 4 3 2 3 3 4 4 4 2 3 4 3 1 3 4 4 2 1 1 1 1
##  [815] 1 1 1 1 1 2 1 1 1 2 2 2 1 2 1 1 4 3 2 3 2 1 2 2 1 3 1 2 1 4 1 2 4 1 2 1 1
##  [852] 1 1 2 1 1 2 1 1 2 1 2 3 2 2 1 3 1 1 1 1 1 2 2 2 2 2 2 3 1 1 1 2 3 4 3 2 3
##  [889] 2 2 3 1 2 2 1 2 4 4 4 2 2 2 2 1 2 3 3 3 2 1 2 1 2 1 1 2 1 1 1 2 1 2 1 2 2
##  [926] 1 2 1 1 1 3 2 3 1 3 2 1 4 2 1 1 2 1 1 1 2 1 2 1 1 2 1 1 1 1 1 1 2 2 1 2 1
##  [963] 1 4 1 1 1 2 2 3 3 3 4 3 1 3 1 3 1 2 1 4 3 1 2 1 1 1 1 1 1 1 1 2 4 4 2 3 3
## [1000] 4 3 4 3 4 3 3 3 2 4 2 4 2 3 4 2 2 2 4 4 3 2 4 4 4 3 2 4 3 3 4 2 2 4 4 2 2
## [1037] 4 1 4 1 4 2 2 1 2 1 2 2 1 2 2 1 2 1 1 3 3 1 2 1 1 2 1 1 1 2 1 2 1 1 3 3 2
## [1074] 3 2 2 1 3 2 3 4 3 1 3 1 2 2 1 3 1 2 1 2 2 1 2 2 3 2 4 4 1 2 2 4 2 1 1 1 1
## [1111] 4 4 2 3 2 4 4 3 3 1 2 2 2 1 1 2 1 2 2 2 2 1 1 1 1 1 1 2 1 1 1 2 2 1 2 2 3
## [1148] 1 2 1 4 3 4 4 4 3 4 2 4 3 3 4 3 1 1 1 3 1 1 1 3 3 3 2 1 2 3 3 2 1 1 2 1 4
## [1185] 3 4 3 3 3 2 4 2 3 2 4 1 3 1 1 1 1 1 2 2 2 1
##
## Within cluster sum of squares by cluster:
## [1] 119.6462 156.7847 210.1404 170.6491
##  (between_SS / total_SS =  72.7 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```r
# Save the cluster number in the dataset as column 'cluster_results'
FH2T$pre_cluster_results <- as.factor(pre_cluster$cluster)
```

Visualizing clusters

```r
# Calculate centroids from your K-means result
centroids <- as.data.frame(pre_cluster$centers)
```

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

# Visualize the data with ggplot
library(ggplot2)
ggplot(FH2T, aes(PreMA_z, PreMP_z)) +
  geom_jitter(aes(color = factor(pre_cluster_groups))) +
  geom_point(data = centroids, aes(x = PreMA_z, y = PreMP_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

Characteristic	**lMP_hMA** N = 343[1]	**lMP_lMA** N = 375[1]	**hMP_**
Gender			
Female	198 (58%)	131 (35%)	
Male	145 (42%)	244 (65%)	
Unknown	0 (0%)	0 (0%)	
race_ethnicity			
American Indian/Alaska Native	3 (0.9%)	1 (0.3%)	
Asian	21 (6.3%)	38 (10%)	

Black/African American	18 (5.4%)	18 (4.8%)	
Hispanic/Latino	80 (24%)	72 (19%)	
Native Hawaiian or Other Pacific Islander	0 (0%)	0 (0%)	
Two or more races	7 (2.1%)	13 (3.5%)	
White	207 (62%)	230 (62%)	
Unknown	7	3	
IEP			
0	298 (87%)	312 (83%)	
1	45 (13%)	63 (17%)	
EIP			
0	312 (91%)	336 (90%)	
1	31 (9.0%)	39 (10%)	
GIFTED			
0	328 (96%)	353 (94%)	
1	15 (4.4%)	22 (5.9%)	
ESOL			
0	290 (85%)	326 (87%)	
1	53 (15%)	49 (13%)	
PreMP	3 (2, 4)	3 (2, 4)	
PreMA	19 (17, 22)	10 (7, 12)	
PreMSE	2.80 (2.20, 3.60)	3.60 (3.00, 4.20)	4.4
Unknown	0	3	

[1] n (%); Median (Q1, Q3)

```
## # A tibble: 4 x 5
##   pre_cluster_groups PreMP_mean PreMA_mean PreMP_sd PreMA_sd
##   <fct>                   <dbl>      <dbl>    <dbl>    <dbl>
## 1 lMP_hMA                  2.75      19.6      1.18     3.24
## 2 lMP_lMA                  3.15       9.56     1.38     3.19
## 3 hMP_lMA                  8.22       7.19     1.26     3.25
## 4 hMP_hMA                  6.73      17.1      1.42     3.31
```

```
# Checking normality - normally distributed
FH2T %>%
  group_by(pre_cluster_groups) %>%
  summarise(shapiro_statistic = shapiro.test(PreMP)$statistic,
            p.value = shapiro.test(PreMP)$p.value)
```

Comparison by MP

```
## # A tibble: 4 x 3
##   pre_cluster_groups shapiro_statistic  p.value
##   <fct>                          <dbl>    <dbl>
## 1 lMP_hMA                        0.918 1.02e-12
## 2 lMP_lMA                        0.917 1.76e-13
```

```
## 3 hMP_lMA                          0.908 1.21e-10
## 4 hMP_hMA                          0.900 3.87e-12
```

```r
# Checking homogeneity of variance - not normally distributed
leveneTest(PreMP ~ pre_cluster_groups, data = FH2T)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value   Pr(>F)
## group    3  4.5078 0.003752 **
##       1202
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
bartlett.test(PreMP ~ pre_cluster_groups, data = FH2T)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  PreMP by pre_cluster_groups
## Bartlett's K-squared = 12.749, df = 3, p-value = 0.005212
```

```r
## MP comparison via Dunn test, as variances not normally distributed
dunn.test(FH2T$PreMP, g=FH2T$pre_cluster_groups, method='bonferroni')
```

```
##   Kruskal-Wallis rank sum test
##
## data: x and group
## Kruskal-Wallis chi-squared = 884.7216, df = 3, p-value = 0
##
##
##                              Comparison of x by group
##                                    (Bonferroni)
## Col Mean-|
## Row Mean |    hMP_hMA    hMP_lMA    lMP_hMA
## ---------+---------------------------------
##  hMP_lMA |  -4.588898
##          |    0.0000*
##          |
##  lMP_hMA |   19.80806   23.88337
##          |    0.0000*    0.0000*
##          |
##  lMP_lMA |   17.72150   21.93966  -2.656532
##          |    0.0000*    0.0000*    0.0237*
##
## alpha = 0.05
## Reject Ho if p <= alpha/2
```

```r
# Checking normality - not normally distributed
FH2T %>%
```

```r
  group_by(pre_cluster_groups) %>%
  summarise(shapiro_statistic = shapiro.test(PreMA)$statistic,
            p.value = shapiro.test(PreMA)$p.value)
```

Comparison by MA

```
## # A tibble: 4 x 3
##   pre_cluster_groups shapiro_statistic  p.value
##   <fct>                          <dbl>    <dbl>
## 1 lMP_hMA                        0.953 5.13e- 9
## 2 lMP_lMA                        0.944 1.15e-10
## 3 hMP_lMA                        0.977 1.04e- 3
## 4 hMP_hMA                        0.959 9.79e- 7
```

```r
# Checking homogeneity of variance - normally distributed
leveneTest(PreMA ~ pre_cluster_groups, data = FH2T)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value Pr(>F)
## group    3  0.0547 0.9832
##       1202
```

```r
bartlett.test(PreMA ~ pre_cluster_groups, data = FH2T)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  PreMA by pre_cluster_groups
## Bartlett's K-squared = 0.43606, df = 3, p-value = 0.9327
```

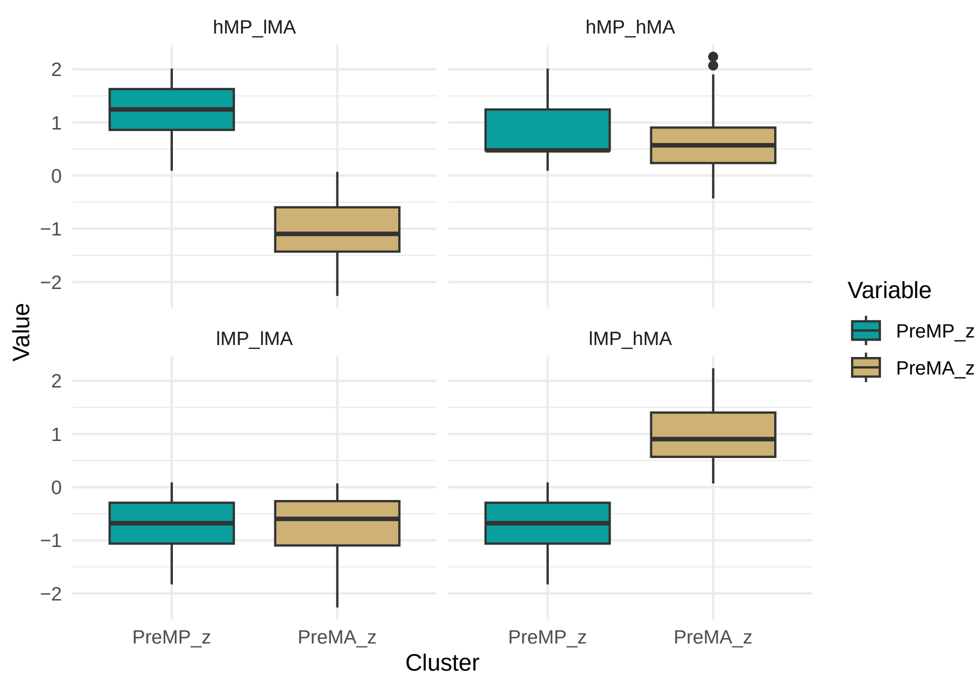
```r
## MA comparison via Dunn test, as data is not normally distributed
dunn.test(FH2T$PreMA, g=FH2T$pre_cluster_groups, method='bonferroni')
```

```
##   Kruskal-Wallis rank sum test
##
## data: x and group
## Kruskal-Wallis chi-squared = 913.3897, df = 3, p-value = 0
##
##
##                            Comparison of x by group
##                                  (Bonferroni)
## Col Mean-|
## Row Mean |    hMP_hMA    hMP_lMA    lMP_hMA
## ---------+---------------------------------
##  hMP_lMA |   18.75219
##          |    0.0000*
##          |
##  lMP_hMA |  -4.841687  -24.53880
##          |    0.0000*    0.0000*
##          |
##  lMP_lMA |   16.57079  -4.353373   23.20161
```

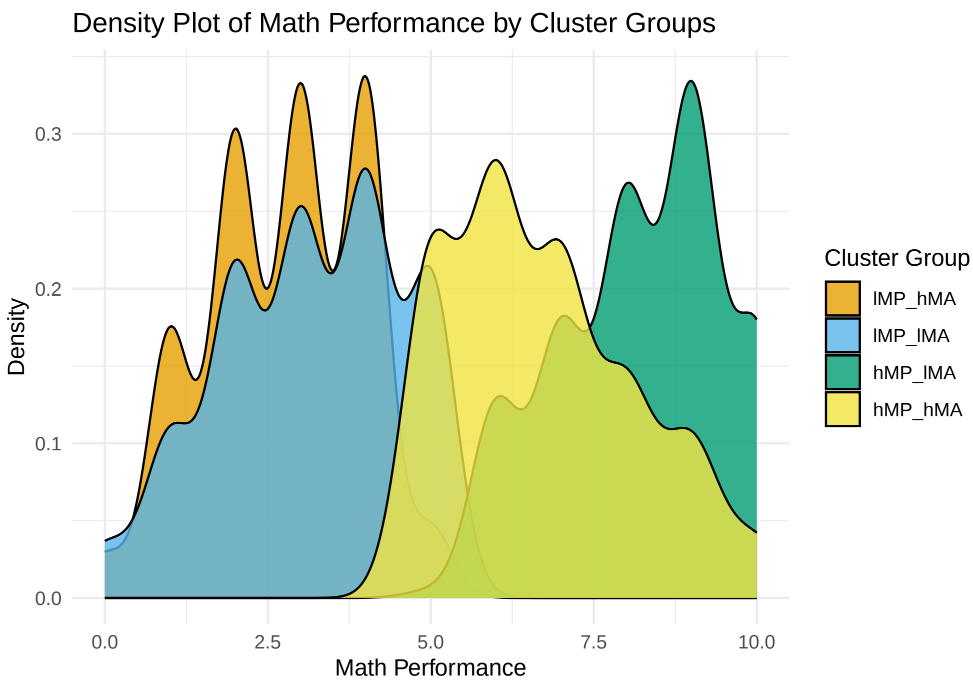
```
##          |    0.0000*    0.0000*    0.0000*
##
## alpha = 0.05
## Reject Ho if p <= alpha/2
```

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

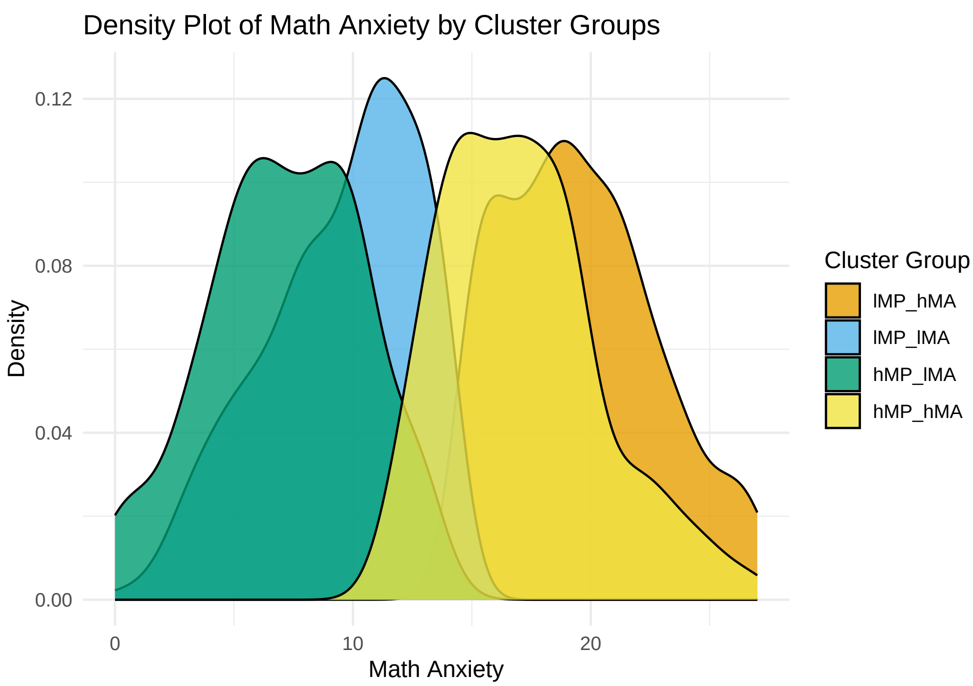
```
## Visualization for both

# Creating long format table
FH2T_data_long <- pivot_longer(FH2T,
                               cols = c('PreMP_z', 'PreMA_z'),
                               names_to = 'Variable',
                               values_to = 'Value')

# Specify levels for factor "Variable" (so MP goes first on the visualization)
FH2T_data_long$Variable <- factor(FH2T_data_long$Variable , levels=c("PreMP_z", "PreMA_z"))


# Create a boxplot for each variable with facets for clusters
ggplot(FH2T_data_long, aes(x = Variable , y = Value, fill = Variable)) +
  geom_boxplot() +
  labs(x = "Cluster", y = "Value") +
  facet_wrap(~ pre_cluster_groups_best, scales = "fixed") +
  scale_fill_manual(values = wes_colors) +
  theme_minimal()
```



MP distribution



MA distribution



Compare by MSE

```r
# Checking normality - not normally distributed
FH2T %>%
  group_by(pre_cluster_groups) %>%
  summarise(shapiro_statistic = shapiro.test(PreMSE)$statistic,
            p.value = shapiro.test(PreMSE)$p.value)
```

Z-scoring, checking normality and comparing MSE

```
## # A tibble: 4 x 3
##   pre_cluster_groups shapiro_statistic       p.value
##   <fct>                          <dbl>         <dbl>
## 1 lMP_hMA                        0.992 0.0601
## 2 lMP_lMA                        0.970 0.000000503
## 3 hMP_lMA                        0.922 0.00000000141
## 4 hMP_hMA                        0.982 0.00224
```

```r
# Checking homogeneity of variance - not normally distributed
leveneTest(PreMSE ~ pre_cluster_groups, data = FH2T)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value    Pr(>F)
## group    3  12.753 3.308e-08 ***
##       1197
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
bartlett.test(PreMSE ~ pre_cluster_groups, data = FH2T)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  PreMSE by pre_cluster_groups
## Bartlett's K-squared = 46.563, df = 3, p-value = 4.304e-10
```

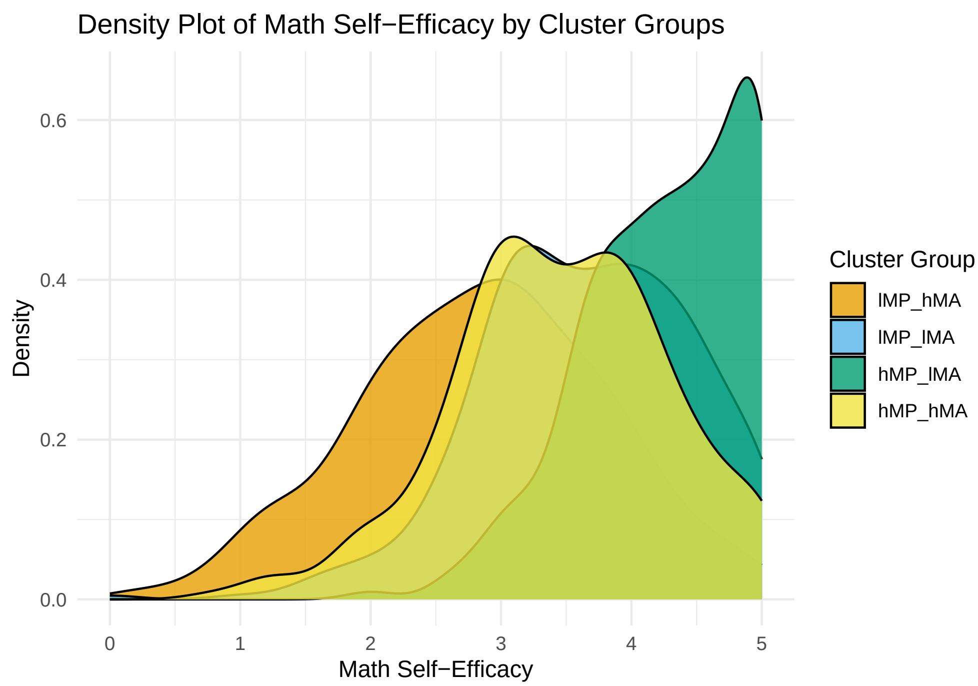
```r
## MA comparison via Dunn test, as data is not normally distributed
dunn.test(FH2T$PreMSE, g=FH2T$pre_cluster_groups, method='bonferroni')
```

```
##   Kruskal-Wallis rank sum test
##
## data: x and group
## Kruskal-Wallis chi-squared = 316.2546, df = 3, p-value = 0
##
##
##                            Comparison of x by group
##                                  (Bonferroni)
## Col Mean-|
## Row Mean |    hMP_hMA    hMP_lMA    lMP_hMA
## ---------+---------------------------------
##  hMP_lMA |  -10.22075
##          |    0.0000*
##          |
```

```
##  lMP_hMA |   6.906861   17.47843
##          |    0.0000*    0.0000*
##          |
##  lMP_lMA |  -2.744899   8.390480  -10.55111
##          |    0.0182*    0.0000*    0.0000*
##
## alpha = 0.05
## Reject Ho if p <= alpha/2
```

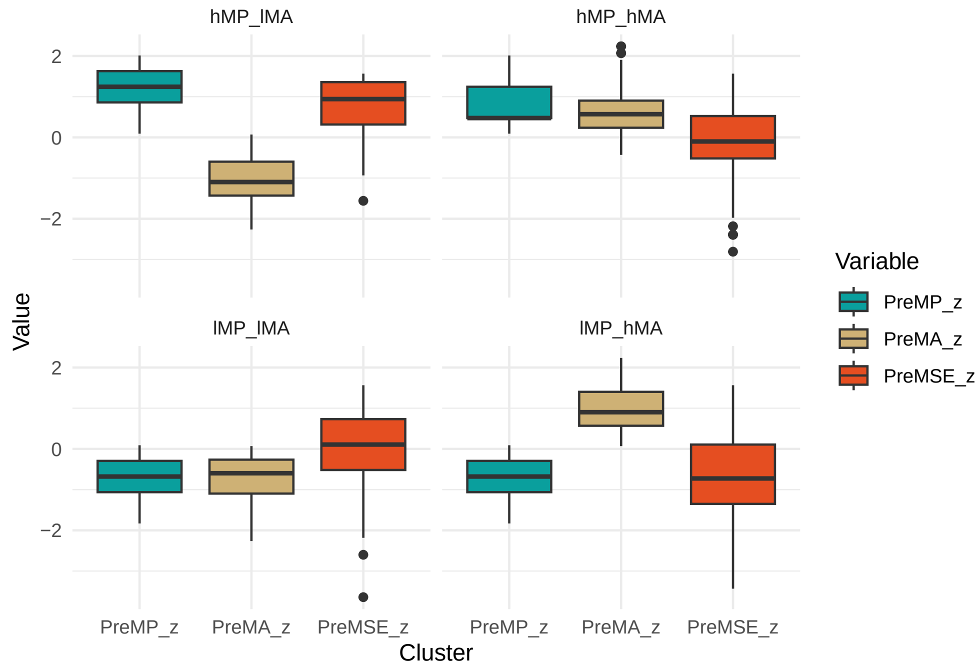
Visualizing MSE distribution

```
## Warning: Removed 5 rows containing non-finite outside the scale range
## (‘stat_density()‘).
```



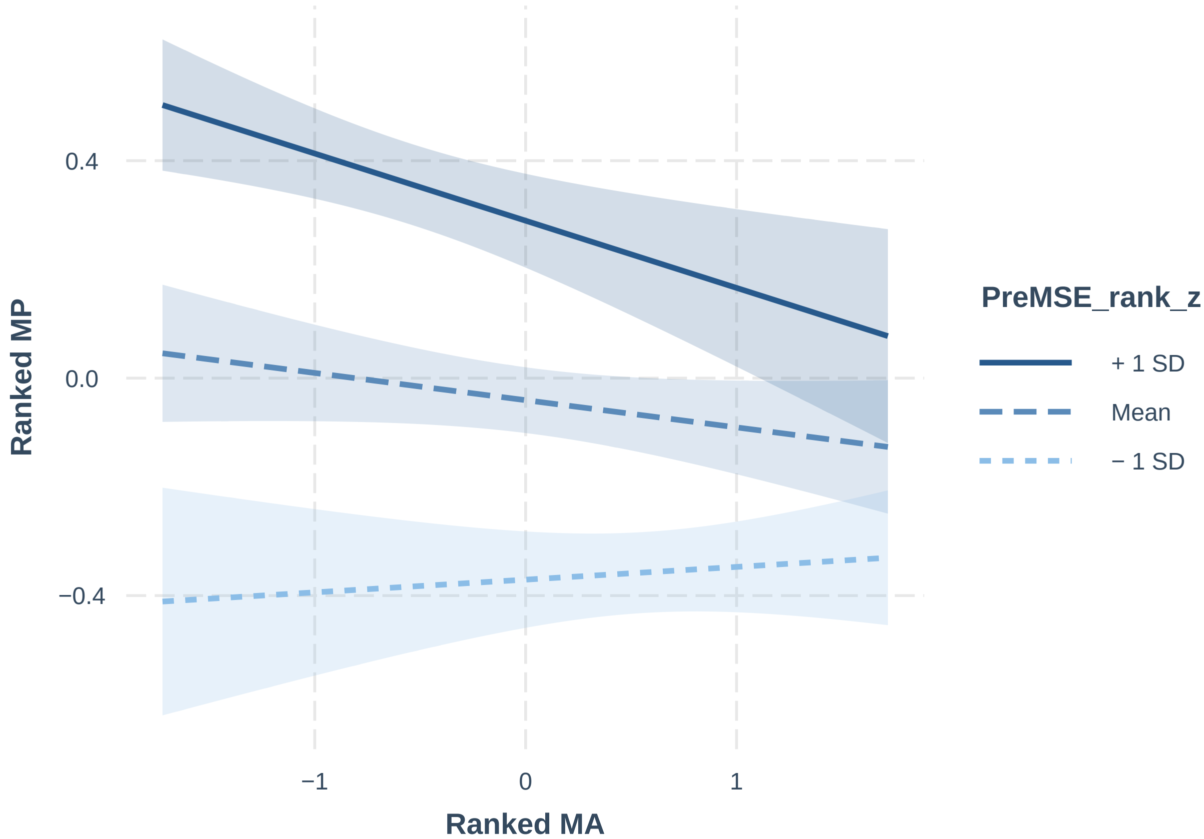
Visualizing with MP and MA within groups

```
## Warning: Removed 5 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```



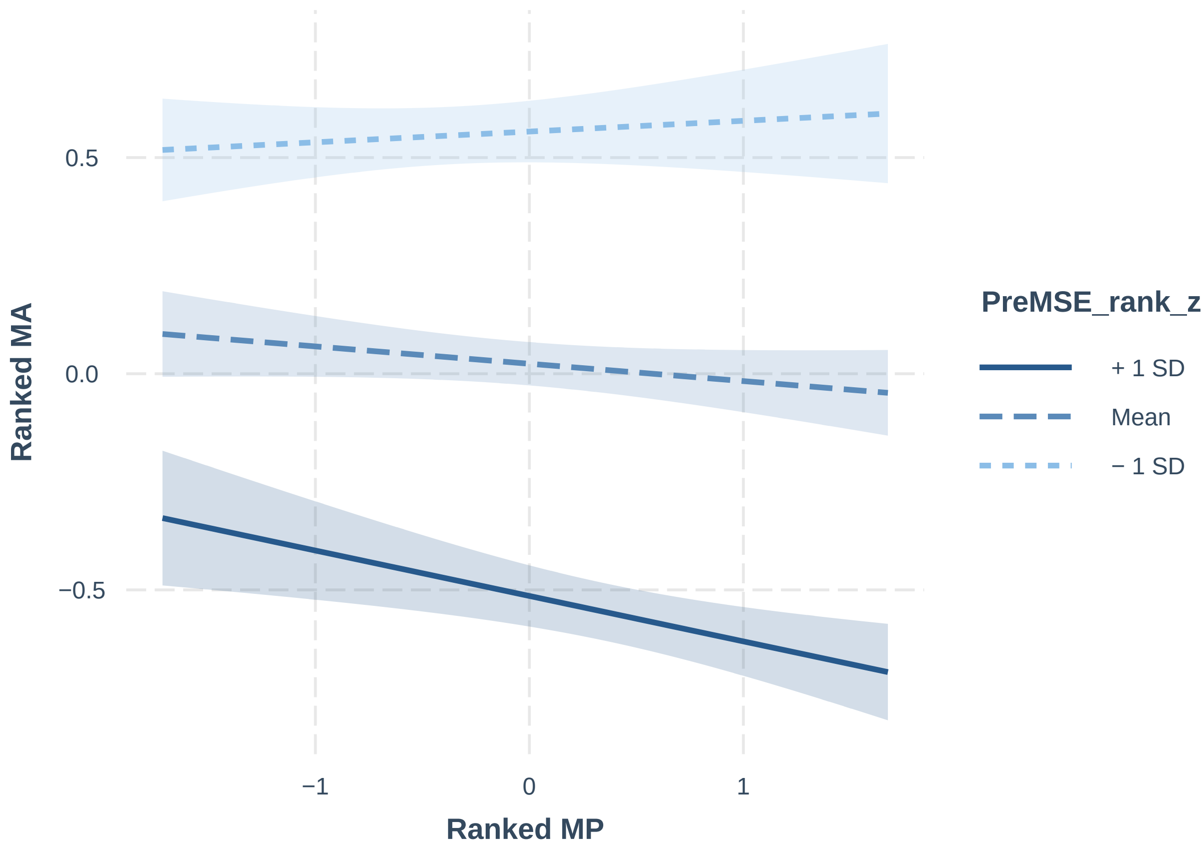
Check self-efficacy as moderator when Math Anxiety predicts Math Performance

```
##
## Call:
## lm(formula = PreMP_rank_z ~ PreMA_rank_z * PreMSE_rank_z, data = ranked_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.32199 -0.75799  0.02009  0.73880  2.16092
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)               -0.04051    0.03083  -1.314  0.18908
## PreMA_rank_z              -0.05009    0.03224  -1.553  0.12058
## PreMSE_rank_z              0.33013    0.03223  10.244  < 2e-16 ***
## PreMA_rank_z:PreMSE_rank_z -0.07349    0.02755  -2.668  0.00774 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9317 on 1202 degrees of freedom
## Multiple R-squared:  0.1341, Adjusted R-squared:  0.1319
## F-statistic: 62.05 on 3 and 1202 DF,  p-value: < 2.2e-16
```



Check self-efficacy as moderator when Math Performance predicts Math Anxiety

```
##
## Call:
## lm(formula = PreMA_rank_z ~ PreMP_rank_z * PreMSE_rank_z, data = ranked_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.23457 -0.63893 -0.05456  0.66925  2.25269
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   0.02312    0.02554   0.905   0.3656
## PreMP_rank_z                 -0.04013    0.02569  -1.562   0.1185
## PreMSE_rank_z                -0.53712    0.02565 -20.944   <2e-16 ***
## PreMP_rank_z:PreMSE_rank_z -0.06496    0.02490  -2.609   0.0092 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8317 on 1202 degrees of freedom
## Multiple R-squared:  0.3099, Adjusted R-squared:  0.3082
## F-statistic:   180 on 3 and 1202 DF,  p-value: < 2.2e-16
```



PreMSE_rank_z
+ 1 SD
Mean
− 1 SD
Ranked MA
0.5
0.0
−0.5
−1
0
1
Ranked MP