MPxMA_Replication

2024-11-05

Sample desctiptives

Correlation matrix

```
##
##
## Table 1
##
## Means, standard deviations, and correlations with confidence intervals
##
##
##   Variable        M     SD   1
##   1. Math_perform 7.86  3.22
##
##   2. Math_anxiety 12.42 6.10 -.27**
##                              [-.35, -.18]
##
##
## 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.
##
```

Demographics

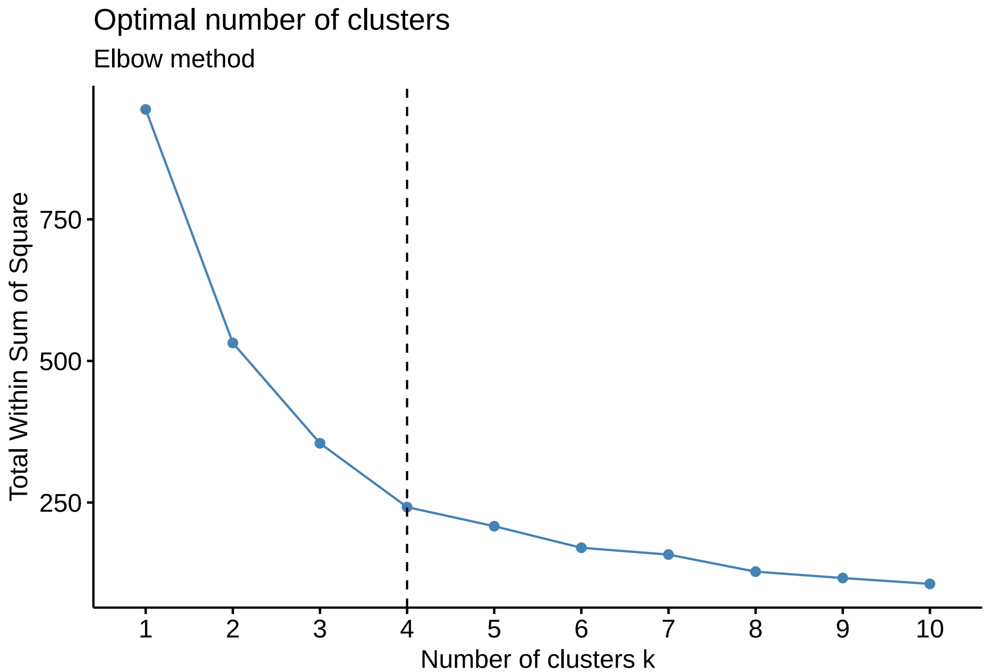
Characteristic	**N = 473**[1]
Gender_Female	219 (49%)
Unknown	22
Gifted	
0	258 (83%)
1	54 (17%)
Unknown	161
ELL	
0	214 (97%)
1	7 (3.2%)
Unknown	252
PRE_SC	8.0 (5.0, 11.0)

MA_TOTAL_SC	12.0 (7.0, 17.0)
Race_Ethnicity	
American Indian	5 (1.1%)
Asian	10 (2.2%)
Black	12 (2.7%)
Hispanic	12 (2.7%)
Multi-racial	22 (4.9%)
Other	5 (1.1%)
White	385 (85%)
Unknown	22

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

Choosing number of clusters

```r
# 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")
```



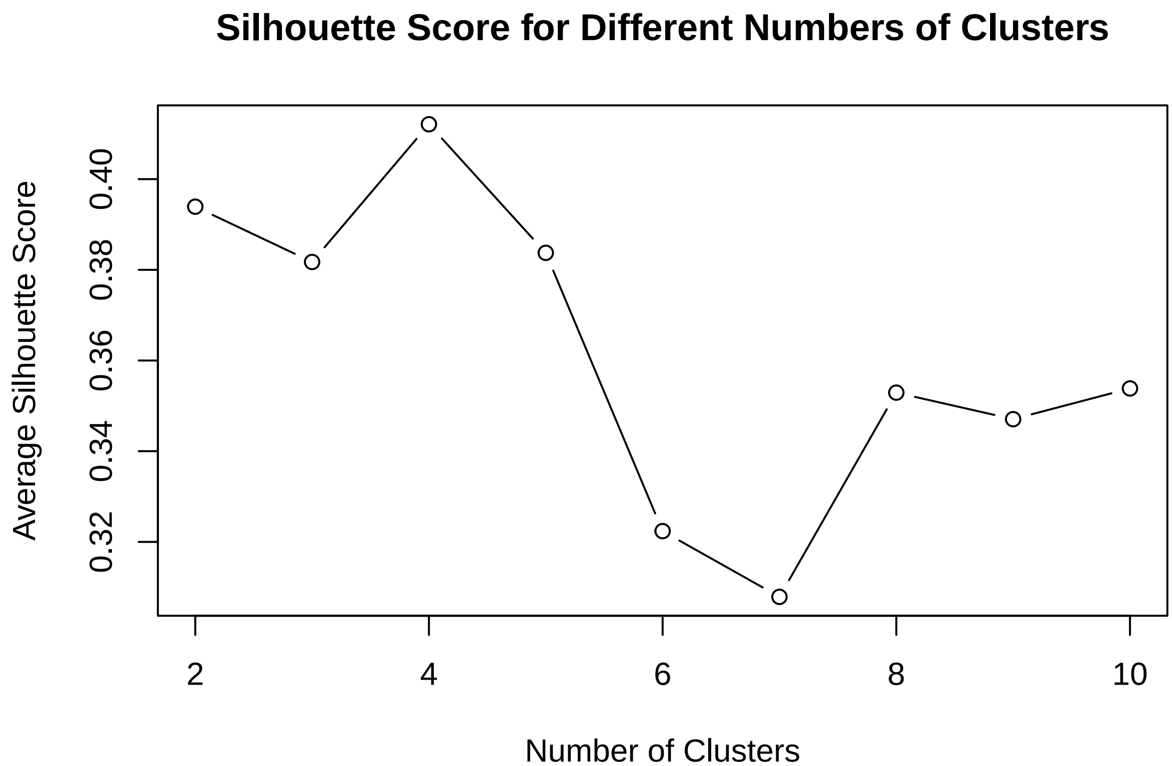
Elbow method

```
# 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_scor
```

Silhouette scores

```
## The optimal number of clusters is 4 with an average Silhouette score of 0.4120973
```

```
# 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

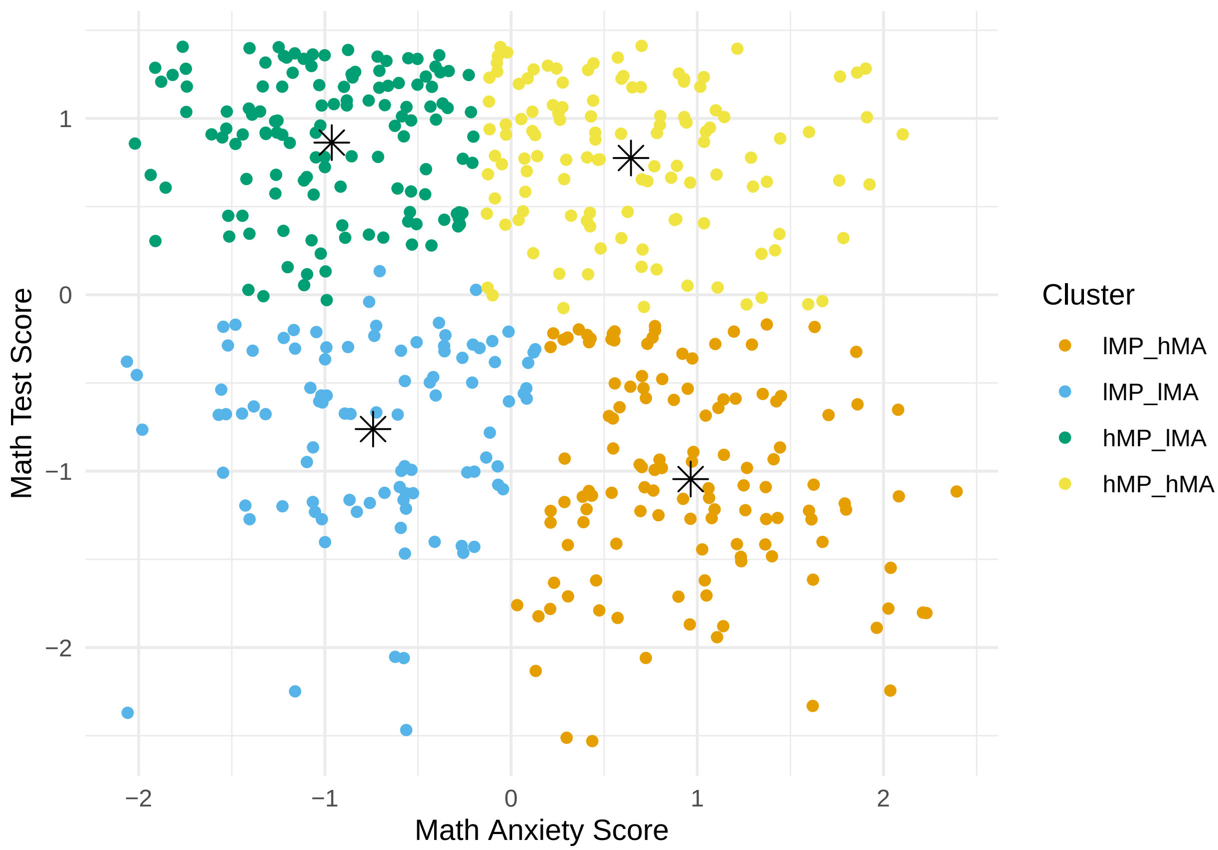
```
### --- Applying k-means clustering
set.seed(20)
cluster <- kmeans(PRE_z, centers = 4, nstart = 25) # put the optimal number of clusters in "centers"
print(cluster)
```

```
## K-means clustering with 4 clusters of sizes 130, 120, 98, 125
##
## Cluster means:
##      PRE_SC_z MA_TOTAL_SC_z
## 1  0.8628612    -0.9628484
## 2  0.7755438     0.6438046
## 3 -0.7613346    -0.7411263
## 4 -1.0450113     0.9643529
##
## Clustering vector:
##   [1] 1 4 1 3 2 4 3 3 3 1 4 2 3 1 4 1 3 1 4 3 4 1 1 4 1 2 3 1 4 3 2 4 2 4 4 1 1
##  [38] 4 4 2 2 1 4 3 2 1 4 2 2 3 4 3 2 1 2 1 3 3 3 3 2 4 4 1 4 4 4 4 4 2 2 1 4 2
##  [75] 3 4 3 2 1 2 2 2 2 1 3 4 2 1 3 2 1 1 2 3 1 2 1 3 1 3 1 4 1 1 1 3 1 3 2 2 3
## [112] 4 2 4 2 2 1 1 4 4 3 2 1 2 3 4 3 4 4 2 1 3 1 4 4 1 2 1 2 3 2 1 1 2 3 4 3 4
## [149] 4 2 4 4 4 2 3 4 2 1 2 4 2 4 1 1 2 1 2 1 4 2 3 1 4 1 3 4 4 4 3 3 3 2 4 3 1
## [186] 3 1 4 1 2 4 2 2 2 1 3 1 1 4 2 3 2 1 4 4 1 4 1 3 1 1 1 3 2 2 3 2 2 2 4 3 3
## [223] 2 1 3 4 2 3 3 2 2 1 3 4 3 4 2 1 1 2 4 2 3 4 4 1 2 3 4 3 3 4 1 4 2 2 4 4 1
```

```
## [260] 1 4 2 2 2 3 2 1 3 1 3 3 2 3 2 3 1 4 1 1 4 4 3 4 4 2 4 2 1 2 1 2 1 3 1 1 1
## [297] 1 2 4 1 4 3 1 3 2 4 2 3 1 3 4 1 4 2 3 1 4 4 1 2 4 2 2 3 4 2 4 3 3 3 1 2 4
## [334] 2 3 3 2 1 1 3 2 3 4 4 3 4 1 2 1 2 3 4 2 3 1 3 1 2 1 2 2 4 3 4 4 4 1 1 1 1
## [371] 2 2 2 1 1 1 2 1 2 1 4 4 4 1 4 1 4 3 2 4 1 2 3 4 3 1 2 4 2 2 4 1 1 3 4 1 1
## [408] 1 4 4 3 1 1 2 1 2 1 4 3 2 2 4 1 4 2 3 3 4 3 2 4 2 4 1 1 2 1 1 2 4 4 1 2 4
## [445] 2 4 2 4 4 1 2 4 2 3 2 3 4 4 3 1 1 4 1 1 3 1 1 3 4 3 4 1 1
##
## Within cluster sum of squares by cluster:
## [1] 47.33279 58.73985 56.86017 78.45883
##  (between_SS / total_SS =  74.4 %)
##
## 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'
data$cluster_results <- as.factor(cluster$cluster)
```

Visualizing clusters



Clusters' demographics

Characteristic	**lMP_hMA** N = 125[1]	**lMP_lMA** N = 98[1]	**hMP_lMA** N = 130[1]	**hMP_h**
Gender_Female	60 (52%)	35 (38%)	50 (40%)	7
Unknown	9	7	4	
Gifted				
0	83 (97%)	53 (84%)	63 (73%)	5
1	3 (3.5%)	10 (16%)	23 (27%)	1
Unknown	39	35	44	
ELL				
0	56 (93%)	48 (98%)	55 (96%)	55
1	4 (6.7%)	1 (2.0%)	2 (3.5%)	
Unknown	65	49	73	
PRE_SC	4.0 (3.0, 6.0)	6.0 (4.0, 7.0)	11.0 (10.0, 12.0)	11.0
MA_TOTAL_SC	18.0 (16.0, 21.0)	9.0 (6.0, 11.0)	6.0 (5.0, 9.0)	16.0

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

```
## # A tibble: 4 x 5
##   cluster_groups PRE_SC_mean MA_TOTAL_SC_mean PRE_SC_sd MA_TOTAL_SC_sd
##   <fct>                <dbl>            <dbl>     <dbl>          <dbl>
## 1 lMP_hMA               4.50            18.3       1.86           3.32
## 2 lMP_lMA               5.41             7.90      1.73           3.31
## 3 hMP_lMA              10.6              6.55      1.25           2.83
## 4 hMP_hMA              10.4             16.3       1.33           3.47
```

Comparison by MP

```
## # A tibble: 4 x 3
##   cluster_groups shapiro_statistic       p.value
##   <fct>                      <dbl>         <dbl>
## 1 lMP_hMA                    0.927 0.00000450
## 2 lMP_lMA                    0.878 0.000000196
## 3 hMP_lMA                    0.863 0.00000000130
## 4 hMP_hMA                    0.889 0.0000000567
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value    Pr(>F)
## group   3  6.0623 0.0004696 ***
##       469
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  PRE_SC by cluster_groups
## Bartlett's K-squared = 27.625, df = 3, p-value = 4.354e-06
```

```
##   Kruskal-Wallis rank sum test
```

```
##
## data: x and group
## Kruskal-Wallis chi-squared = 360.2688, df = 3, p-value = 0
##
##
##                            Comparison of x by group
##                                  (Bonferroni)
## Col Mean-|
## Row Mean |    hMP_hMA    hMP_lMA    lMP_hMA
## ---------+---------------------------------
##  hMP_lMA |  -0.872491
##          |     1.0000
##          |
##  lMP_hMA |   13.98125   15.14567
##          |    0.0000*    0.0000*
##          |
##  lMP_lMA |   11.36058   12.38788  -1.779265
##          |    0.0000*    0.0000*     0.2256
##
## alpha = 0.05
## Reject Ho if p <= alpha/2
```

Comparison by MA

```
## # A tibble: 4 x 3
##   cluster_groups shapiro_statistic   p.value
##   <fct>                      <dbl>     <dbl>
## 1 lMP_hMA                    0.953 0.000256
## 2 lMP_lMA                    0.954 0.00175
## 3 hMP_lMA                    0.959 0.000618
## 4 hMP_hMA                    0.930 0.0000100
```

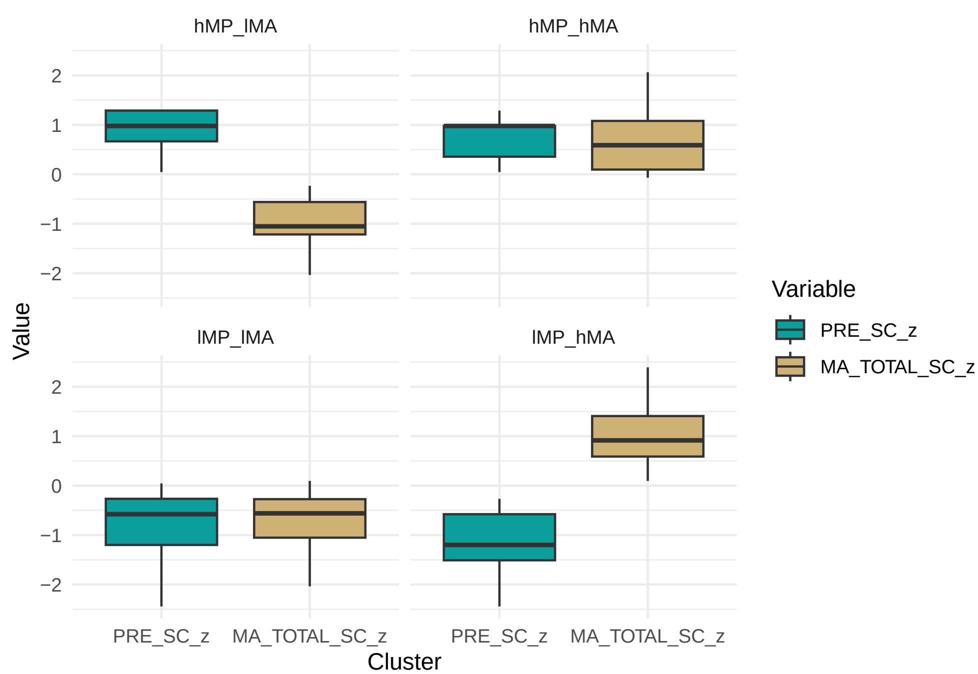
```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value Pr(>F)
## group   3  1.6852 0.1693
##       469
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  MA_TOTAL_SC by cluster_groups
## Bartlett's K-squared = 5.718, df = 3, p-value = 0.1262
```

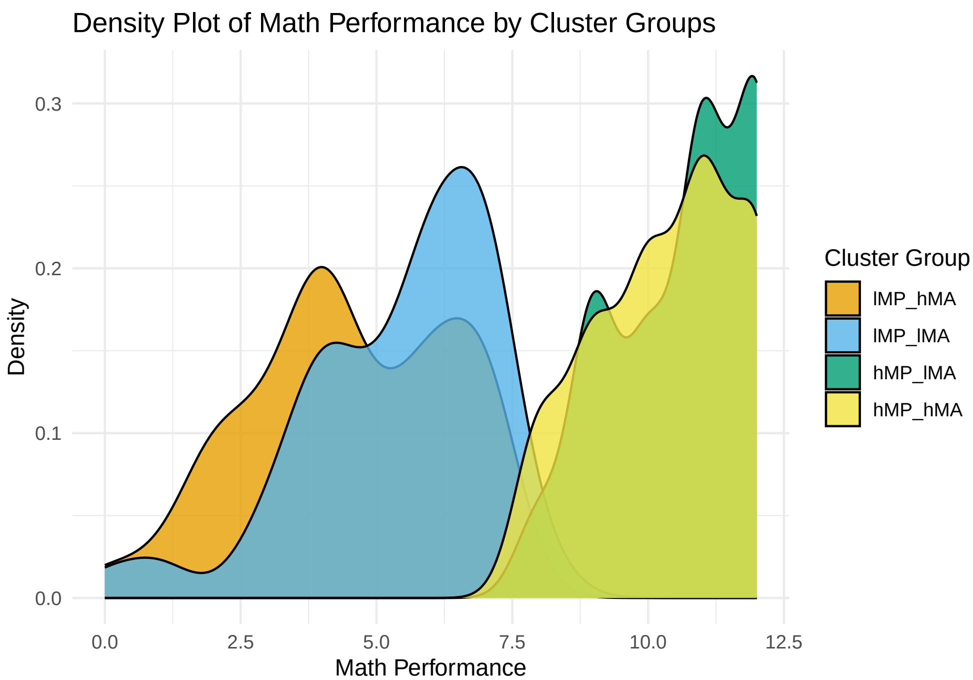
```
##   Kruskal-Wallis rank sum test
##
## data: x and group
## Kruskal-Wallis chi-squared = 357.0305, df = 3, p-value = 0
##
##
##                            Comparison of x by group
##                                  (Bonferroni)
## Col Mean-|
## Row Mean |    hMP_hMA    hMP_lMA    lMP_hMA
```

```
## ---------+--------------------------------
##  hMP_lMA |   13.10723
##          |    0.0000*
##          |
##  lMP_hMA |  -2.400659  -15.69491
##          |     0.0491    0.0000*
##          |
##  lMP_lMA |   10.50601  -1.710750   12.87573
##          |    0.0000*     0.2614    0.0000*
##
## alpha = 0.05
## Reject Ho if p <= alpha/2
```

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



MP distribution



MA distribution