Knowledge & Motivation Instrument Correlations

Knowledge & Motivation Instrument Correlations

This notebook explores data from a multi-survey study investigating the relationships between sustainable behaviors, knowledge, and attitudes. The study involved participants completing five different surveys, each designed to measure different aspects of these constructs. This analysis focuses on understanding these relationships, with a particular emphasis on the connection between knowledge and motivation.

Data and Survey Instruments

The dataset combines responses from five different surveys into a single data frame, structured such that each row represents a participant, and each column represents a response, or a derived score. The five underlying surveys are:

- **Energy Literacy Survey (ELS01-ELS08):** Assesses participants' knowledge of energy concepts through multiple-choice questions.
- **Attari Energy Survey - Part 1:**
  - **Perceived Difficulty Items (ATT01-ATT15):** Measures how easy or hard participants would find it to adopt energy-saving behaviors, using a rating scale.
  - **Numeracy Questions (ATT16-ATT18):** Assesses numerical literacy through probability questions requiring numeric answers.
- **Attari Energy Survey - Part 2:**
  - **Relative Energy Usage (ATT19-ATT27):** Asks participants to estimate the relative energy usage of various devices compared to a 100-Watt bulb, using a numeric response format.
  - **Relative Energy Savings (ATT28-ATT33):** Asks participants to estimate the relative energy savings of various actions compared to turning off a 100-Watt bulb, using a numeric response format.
- **Recycling Study Questions (RS01-RS06):** Assesses participants' attitudes towards the environment and politics.

Exploratory Analysis of Multi-Survey Study on Sustainable Behaviors

This notebook presents an exploratory analysis of response data from a multi-survey study focused on understanding the relationships between sustainable behaviors, knowledge, and attitudes. The study involved participants completing five different surveys, each designed to measure different aspects of these constructs. This analysis aims to examine the relationships between the different survey instruments, with a particular focus on the connection between knowledge and motivation.

Data Loading and Preparation

This code block loads the required R libraries for data analysis and visualization. It then reads the survey response data from two RDS files, `draw.rds` and `dinst.rds`, which presumably contain responses from different groups or conditions. Finally, it combines subsets of the data corresponding to different surveys (Attari Energy Survey Part 1 and Part 2, Energy Literacy Survey, and Recycling Study) into separate data frames.

```r
pacman::p_load(dplyr,purrr,tidyr,here, haven,tibble,ggplot2,ggh4x, patchwork,
    lme4,knitr,kableExtra,gt,pander,flextable,ggh4x,psych,corrplot,factoextra)
options(digits=2, scipen=999, dplyr.summarise.inform=FALSE)

library(gridExtra)
library(factoextra)
library(mgcv)
library(lavaan)
library(CCA)
library(qgraph)
library(rpart)
library(rpart.plot)
library(mclust)
library(tidyLPA)



select = dplyr::select

source(here("scripts","survey_functions.R"))

# Load data from RDS files
draw <- readRDS(here("data", "draw.rds"))
#dinst <- readRDS(here("data", "dinst.rds"))

# Combine data from different sources
aes1 <- draw |> select(id, ATT01:ATT18)
#aes2 <- dinst |> select(id, ATT01:ATT18)
aes_combined <- bind_rows(aes1)

att_useSave <- draw |> select(id, ATT19:ATT33)
#att_useSave2 <- dinst |> select(id, ATT19:ATT33)
att2_combined <- bind_rows(att_useSave)

els1 <- draw |> select(id, ELS01:ELS08)
```

```
# els2 <- dinst |> select(id, ELS01:ELS08)
els <- bind_rows(els1)

rs1 <- draw |> select(id, RS01:RS06)
#rs2 <- dinst |> select(id, RS01:RS06)
rs <- bind_rows(rs1)
```

This code block processes the raw survey responses to generate meaningful scores for each participant. It utilizes custom functions (`analyze_attari_survey_part1`, `analyze_attari_survey`, `analyze_els_survey`, `analyze_recycling_survey`) to calculate scores based on the specific coding schemes of each survey. These individual scores are then combined into a single data frame `combined_scores`, where each row represents a participant and each column represents a score from one of the surveys. Finally, the columns are renamed for better readability.

Data Summarization and Scoring

```
# Analyze and score each survey
attari1 <- analyze_attari_survey_part1(aes_combined)
attari2_scores <- analyze_attari_survey(att2_combined)
els_scores <- analyze_els_survey(els)
rs_scores <- analyze_recycling_survey(rs)

# Combine all scores into one dataframe
combined_scores <- attari1 %>%
  left_join(attari2_scores, by = "id") %>%
  left_join(els_scores, by = "id") %>%
  left_join(rs_scores, by = "id")

# Rename columns for clarity
names(combined_scores) <- c(
  "id", "perceived_difficulty", "numeracy",
  "energy_use", "energy_save",
  "els_accuracy", "els_score",
  "env_attitude", "env_attitude_z",
  "pol_conservatism", "pol_conservatism_z"
)
```

Preliminary Data Exploration

```
# Preview the combined scores data
combined_scores |> head(5) |> kable() |> kable_styling("striped", full_width = F)
```

id	perceived_difficulty	numeracy	energy_use	energy_save	els_accuracy	els_score	env_attitude	env_attitude_z
1	0.51	0.98	0.7601	0.78	6	0.79	3.2	-0.45
2	-0.48	0.98	0.0015	-0.43	5	0.27	3.5	-0.12
3	1.90	-1.87	-1.2393	0.53	4	-0.25	3.0	-0.77
4	-0.70	-1.26	0.9531	1.73	2	-1.29	3.8	0.21
5	0.79	0.98	-0.0468	-0.48	3	-0.77	3.8	0.21

This code provides a glimpse into the structure and content of the `combined_scores` data frame.

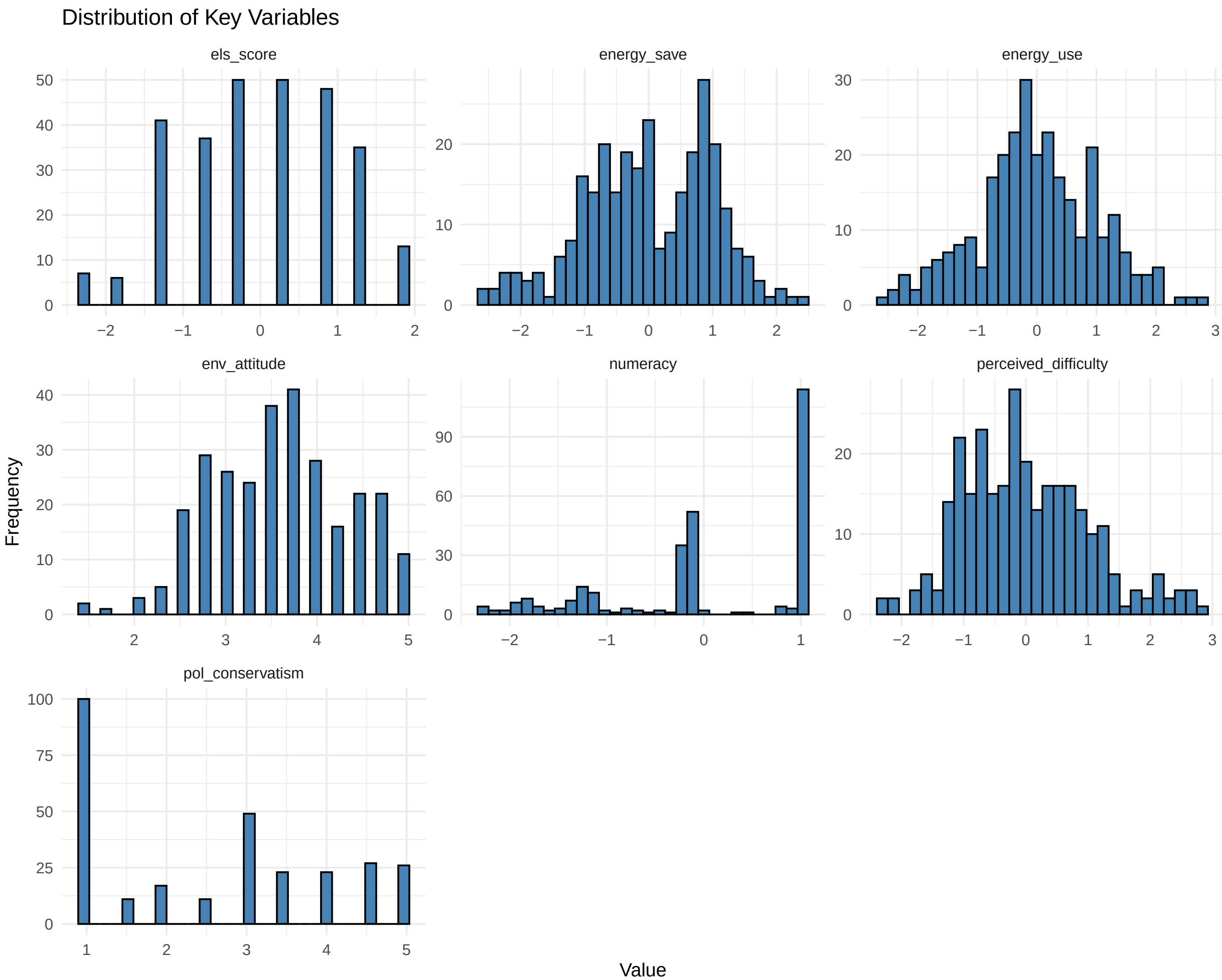
id	perceived_difficulty	numeracy	energy_use	energy_save	els_accuracy	els_score	env_attitude	env_attitude_z
1	0.61	1.5	1.101	1.01	6	0.74	3.2	-0.43
2	-0.45	1.5	0.137	-0.46	5	0.20	3.5	-0.11
3	2.09	-2.0	-1.440	0.70	4	-0.33	3.0	-0.76
4	-0.69	-1.3	1.346	2.16	2	-1.40	3.8	0.22
5	0.91	1.5	0.075	-0.52	3	-0.87	3.8	0.22

This table shows the first five rows of the `combined_scores` data frame, providing a snapshot of the calculated scores for each participant across the different measures. The scores include perceived difficulty, numeracy, energy use, energy save, ELS accuracy, ELS score, environmental attitude, and the z-score of environmental attitude.

Descriptive Statistics

```
# Histograms of key variables
key_vars <- combined_scores %>%
  select(perceived_difficulty, numeracy, energy_use, energy_save, els_score, env_attitude, pol_conservatism)

# Melt the data for plotting
melted_vars <- key_vars %>%
  gather(key = "variable", value = "value")

# Plot histograms
ggplot(melted_vars, aes(x = value)) +
  geom_histogram(bins = 30, fill = "steelblue", color = "black") +
```

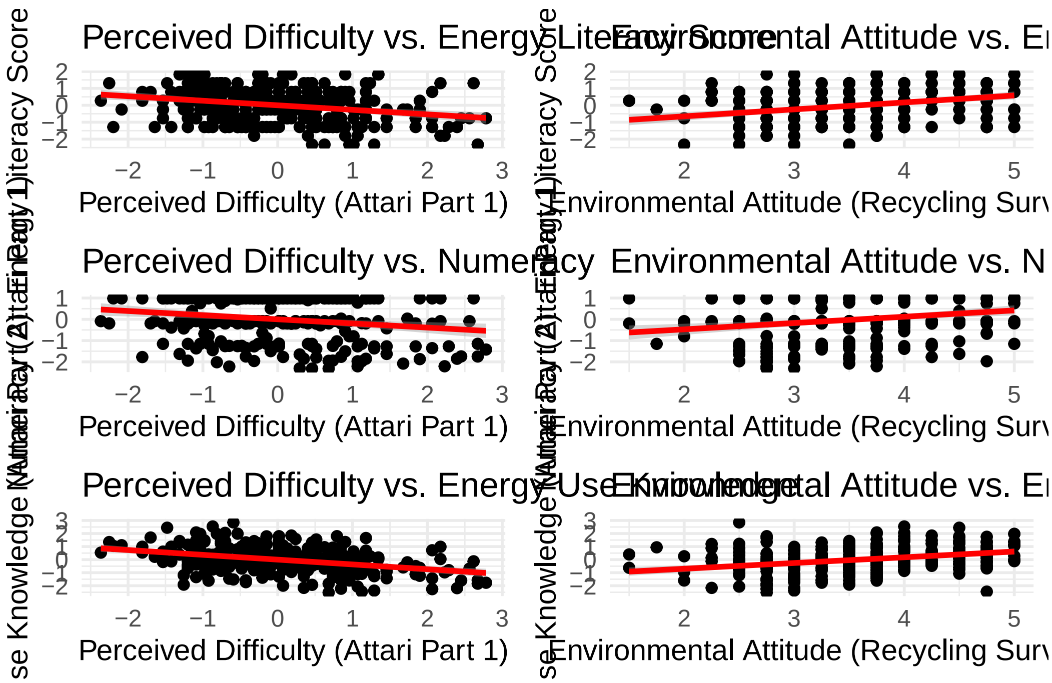
```r
  facet_wrap(~variable, scales = "free") +
  theme_minimal() +
  labs(title = "Distribution of Key Variables", x = "Value", y = "Frequency")
```




```r
# Scatter plot of perceived difficulty vs. ELS score
plot_pd_els <- ggplot(combined_scores, aes(x = perceived_difficulty, y = els_score)) +
  geom_point() +
  geom_smooth(method = "lm", se = TRUE, color = "red") + # Add linear regression line
  labs(title = "Perceived Difficulty vs. Energy Literacy Score",
       x = "Perceived Difficulty (Attari Part 1)",
       y = "Energy Literacy Score (ELS)") +
  theme_minimal()


# Scatter plot of environmental attitude vs. ELS score
```

```
plot_ea_els <- ggplot(combined_scores, aes(x = env_attitude, y = els_score)) +
  geom_point() +
  geom_smooth(method = "lm", se = TRUE, color = "red") + # Add linear regression line
  labs(title = "Environmental Attitude vs. Energy Literacy Score",
       x = "Environmental Attitude (Recycling Survey)",
       y = "Energy Literacy Score (ELS)") +
  theme_minimal()

# Scatter plot of perceived difficulty vs. Numeracy
plot_pd_num <- ggplot(combined_scores, aes(x = perceived_difficulty, y = numeracy)) +
  geom_point() +
  geom_smooth(method = "lm", se = TRUE, color = "red") + # Add linear regression line
  labs(title = "Perceived Difficulty vs. Numeracy",
       x = "Perceived Difficulty (Attari Part 1)",
       y = "Numeracy (Attari Part 1)") +
  theme_minimal()

# Scatter plot of environmental attitude vs. Numeracy
plot_ea_num <- ggplot(combined_scores, aes(x = env_attitude, y = numeracy)) +
  geom_point() +
  geom_smooth(method = "lm", se = TRUE, color = "red") + # Add linear regression line
  labs(title = "Environmental Attitude vs. Numeracy",
       x = "Environmental Attitude (Recycling Survey)",
       y = "Numeracy (Attari Part 1)") +
  theme_minimal()

# Scatter plot of perceived difficulty vs. Energy Use Knowledge
plot_pd_eu <- ggplot(combined_scores, aes(x = perceived_difficulty, y = energy_use)) +
  geom_point() +
  geom_smooth(method = "lm", se = TRUE, color = "red") + # Add linear regression line
  labs(title = "Perceived Difficulty vs. Energy Use Knowledge",
       x = "Perceived Difficulty (Attari Part 1)",
       y = "Energy Use Knowledge (Attari Part 2)") +
  theme_minimal()

# Scatter plot of environmental attitude vs. Energy Use Knowledge
plot_ea_eu <- ggplot(combined_scores, aes(x = env_attitude, y = energy_use)) +
  geom_point() +
  geom_smooth(method = "lm", se = TRUE, color = "red") + # Add linear regression line
```

```
  labs(title = "Environmental Attitude vs. Energy Use Knowledge",
       x = "Environmental Attitude (Recycling Survey)",
       y = "Energy Use Knowledge (Attari Part 2)") +
  theme_minimal()


# Arrange and display the plots (you might need to install gridExtra if you haven't)
gridExtra::grid.arrange(plot_pd_els, plot_ea_els, plot_pd_num, plot_ea_num, plot_pd_eu, plot_ea_eu, ncol = 2)
```




```
# 3. Simple Linear Regression Models

# Model: ELS score predicted by perceived difficulty
model_els_pd <- lm(els_score ~ perceived_difficulty, data = combined_scores)
summary(model_els_pd)
```

```
Call:
lm(formula = els_score ~ perceived_difficulty, data = combined_scores)

Residuals:
    Min      1Q  Median      3Q     Max
-2.1980 -0.7246  0.0506  0.7023  2.1841

Coefficients:
                                  Estimate            Std. Error t value
```

```
(Intercept)          -0.000000000000000187  0.056961632532338199     0.0
perceived_difficulty -0.268417318288949602  0.057061128909530606    -4.7
                     Pr(>|t|)
(Intercept)                 1
perceived_difficulty 0.000004 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.96 on 285 degrees of freedom
Multiple R-squared:  0.072, Adjusted R-squared:  0.0688
F-statistic: 22.1 on 1 and 285 DF,  p-value: 0.00000398
```

```r
# Model: ELS score predicted by environmental attitude
model_els_ea <- lm(els_score ~ env_attitude, data = combined_scores)
summary(model_els_ea)
```

```

Call:
lm(formula = els_score ~ env_attitude, data = combined_scores)

Residuals:
    Min      1Q  Median      3Q     Max
-2.2841 -0.7300 -0.0017  0.7265  2.1682

Coefficients:
             Estimate Std. Error t value    Pr(>|t|)
(Intercept)   -1.4739     0.2708   -5.44 0.000000113 ***
env_attitude   0.4105     0.0738    5.56 0.000000061 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.95 on 285 degrees of freedom
Multiple R-squared:  0.098, Adjusted R-squared:  0.0948
F-statistic:   31 on 1 and 285 DF,  p-value: 0.000000061
```

```r
# Model: Numeracy predicted by perceived difficulty
model_num_pd <- lm(numeracy ~ perceived_difficulty, data = combined_scores)
summary(model_num_pd)
```

```

Call:
```

```
lm(formula = numeracy ~ perceived_difficulty, data = combined_scores)

Residuals:
    Min      1Q  Median      3Q     Max
-2.3475 -0.6219 -0.0402  0.9201  1.4913

Coefficients:
                                 Estimate           Std. Error t value Pr(>|t|)
(Intercept)           0.00000000000000174  0.05799966257793866    0.00  1.00000
perceived_difficulty -0.19472790345158641  0.05810097210943094   -3.35  0.00091

(Intercept)
perceived_difficulty ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.98 on 285 degrees of freedom
Multiple R-squared:  0.0379,    Adjusted R-squared:  0.0345
F-statistic: 11.2 on 1 and 285 DF,  p-value: 0.000912
```

```r
# Model: Numeracy predicted by environmental attitude
model_num_ea <- lm(numeracy ~ env_attitude, data = combined_scores)
summary(model_num_ea)
```

```
Call:
lm(formula = numeracy ~ env_attitude, data = combined_scores)

Residuals:
    Min      1Q  Median      3Q     Max
-2.3159 -0.5468  0.0146  0.9343  1.6018

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   -1.0653     0.2778   -3.84  0.00015 ***
env_attitude   0.2967     0.0757    3.92  0.00011 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.98 on 285 degrees of freedom
Multiple R-squared:  0.0512,    Adjusted R-squared:  0.0478
```

```
F-statistic: 15.4 on 1 and 285 DF,  p-value: 0.000111
```

```r
# Model: Energy use knowledge predicted by perceived difficulty
model_eu_pd <- lm(energy_use ~ perceived_difficulty, data = combined_scores)
summary(model_eu_pd)
```

```

Call:
lm(formula = energy_use ~ perceived_difficulty, data = combined_scores)

Residuals:
   Min     1Q Median     3Q    Max 
-2.379 -0.550 -0.062  0.677  2.633 

Coefficients:
                                Estimate           Std. Error t value
(Intercept)          -0.00000000000000115  0.05499889697000458    0.00
perceived_difficulty -0.36728092446317562  0.05509496498552340   -6.67
                          Pr(>|t|)    
(Intercept)                      1    
perceived_difficulty 0.00000000014 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.93 on 285 degrees of freedom
Multiple R-squared:  0.135, Adjusted R-squared:  0.132 
F-statistic: 44.4 on 1 and 285 DF,  p-value: 0.000000000136
```

```r
# Model: Energy use knowledge predicted by environmental attitude
model_eu_ea <- lm(energy_use ~ env_attitude, data = combined_scores)
summary(model_eu_ea)
```

```

Call:
lm(formula = energy_use ~ env_attitude, data = combined_scores)

Residuals:
   Min     1Q Median     3Q    Max 
-2.960 -0.620 -0.002  0.590  3.328 

Coefficients:
```

```
              Estimate Std. Error t value     Pr(>|t|)
(Intercept)   -1.5744     0.2687   -5.86 0.0000000129 ***
env_attitude   0.4385     0.0732    5.99 0.0000000064 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.94 on 285 degrees of freedom
Multiple R-squared:  0.112, Adjusted R-squared:  0.109
F-statistic: 35.9 on 1 and 285 DF,  p-value: 0.00000000636
```

```r
# 4. Multiple Linear Regression Models

# Model: ELS score predicted by both perceived difficulty and environmental attitude
model_els_pd_ea <- lm(els_score ~ perceived_difficulty + env_attitude, data = combined_scores)
summary(model_els_pd_ea)
```

```
Call:
lm(formula = els_score ~ perceived_difficulty + env_attitude,
    data = combined_scores)

Residuals:
    Min      1Q  Median      3Q     Max
-2.1928 -0.6999 -0.0498  0.7130  2.2329

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)           -1.1347     0.2992   -3.79  0.00018 ***
perceived_difficulty  -0.1598     0.0624   -2.56  0.01101 *
env_attitude           0.3160     0.0819    3.86  0.00014 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.94 on 284 degrees of freedom
Multiple R-squared:  0.118, Adjusted R-squared:  0.112
F-statistic: 19.1 on 2 and 284 DF,  p-value: 0.0000000172
```

```r
# Model: Numeracy predicted by both perceived difficulty and environmental attitude
model_num_pd_ea <- lm(numeracy ~ perceived_difficulty + env_attitude, data = combined_scores)
summary(model_num_pd_ea)
```

```
Call:
lm(formula = numeracy ~ perceived_difficulty + env_attitude,
    data = combined_scores)

Residuals:
    Min      1Q  Median      3Q     Max
-2.1369 -0.6210  0.0337  0.8865  1.6788

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)           -0.8181     0.3086   -2.65   0.0085 **
perceived_difficulty  -0.1164     0.0644   -1.81   0.0717 .
env_attitude           0.2279     0.0844    2.70   0.0074 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.97 on 284 degrees of freedom
Multiple R-squared:  0.062, Adjusted R-squared:  0.0554
F-statistic: 9.38 on 2 and 284 DF,  p-value: 0.000114
```

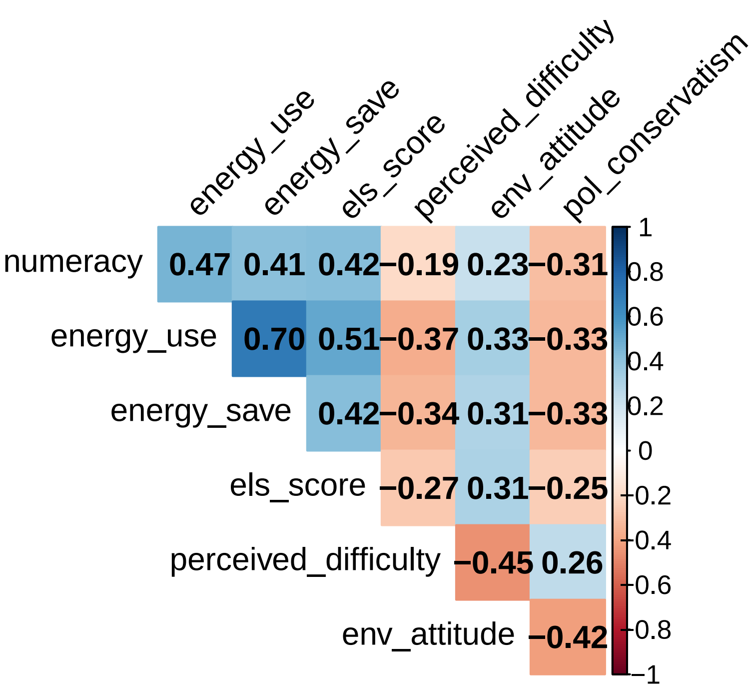
```
# Model: Energy use knowledge predicted by both perceived difficulty and environmental attitude
model_eu_pd_ea <- lm(energy_use ~ perceived_difficulty + env_attitude, data = combined_scores)
summary(model_eu_pd_ea)
```

```
Call:
lm(formula = energy_use ~ perceived_difficulty + env_attitude,
    data = combined_scores)

Residuals:
   Min     1Q Median     3Q    Max
-2.468 -0.594 -0.112  0.650  2.992

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)           -0.9972     0.2902   -3.44  0.00068 ***
perceived_difficulty  -0.2718     0.0605   -4.49  0.00001 ***
env_attitude           0.2777     0.0794    3.50  0.00054 ***
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.91 on 284 degrees of freedom
Multiple R-squared:  0.171, Adjusted R-squared:  0.165
F-statistic: 29.2 on 2 and 284 DF,  p-value: 0.0000000000029
```

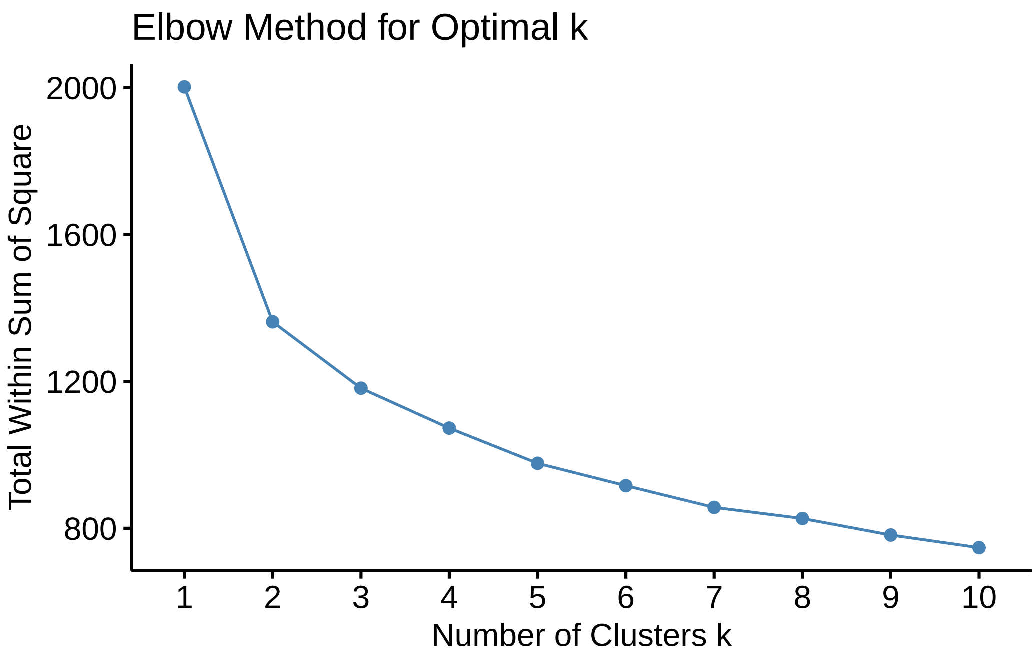
Correlation Analysis

```r
# Select relevant variables
cor_vars <- combined_scores %>%
  select(numeracy, energy_use, energy_save, els_score,
         perceived_difficulty, env_attitude, pol_conservatism)

# Compute correlation matrix
cor_matrix <- cor(cor_vars, use = "pairwise.complete.obs")

# Visualize correlation matrix
corrplot::corrplot(cor_matrix,
                   method = "color",
                   addCoef.col = "black",
                   type = "upper",
                   tl.col  = "black",
                   tl.srt  = 45,
                   diag    = FALSE)
```



Cluster Analysis

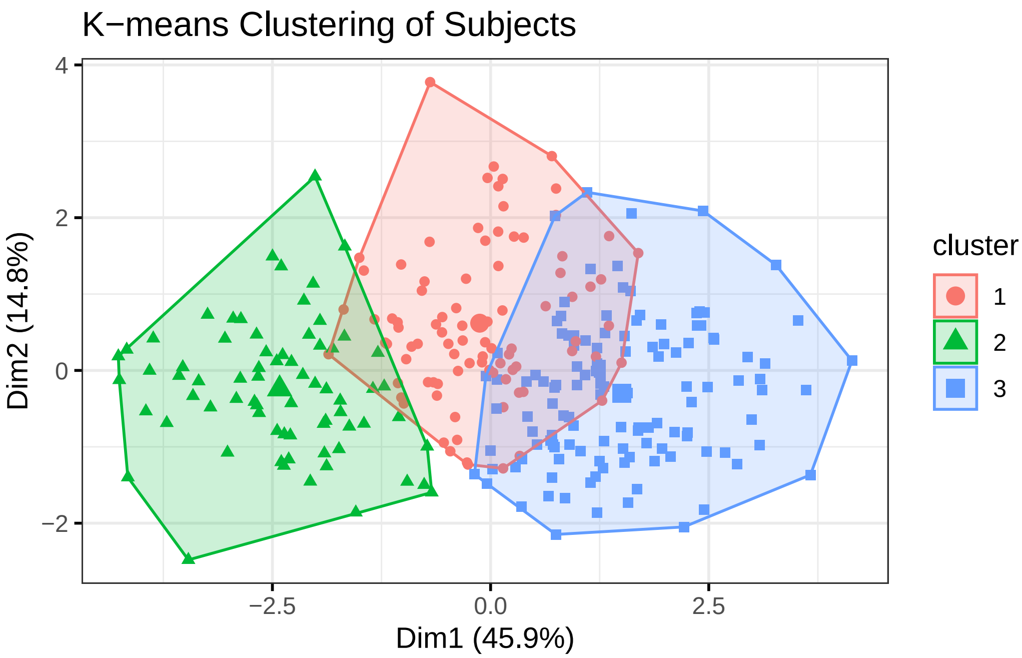
```
# Prepare data for clustering (select relevant variables and scale)
cluster_data <- combined_scores %>%
  select(perceived_difficulty, numeracy, energy_use, energy_save, els_score, env_attitude_z, pol_conservatism_
  na.omit() %>%
  scale()

# Determine optimal number of clusters using the elbow method
fviz_nbclust(cluster_data, kmeans, method = "wss") +
  labs(title = "Elbow Method for Optimal k", x = "Number of Clusters k")
```



This code performs a cluster analysis to identify groups of participants with similar profiles across the measured variables. It first prepares the data by selecting the relevant variables, removing rows with missing values, and scaling the data. Then, it uses the elbow method to determine the optimal number of clusters.

```
# Perform k-means clustering (e.g., with 3 clusters)
set.seed(123)
km_result <- kmeans(cluster_data, centers = 3, nstart = 25)

# Visualize the clusters
fviz_cluster(km_result,
  data = cluster_data,
  geom = "point",
  ellipse.type = "convex",
  ggtheme = theme_bw()
```

```
) +
  labs(title = "K-means Clustering of Subjects")
```



This code performs k-means clustering with 3 clusters (as suggested by the elbow method) and visualizes the clusters using a scatter plot.

Elbow Method for Optimal k

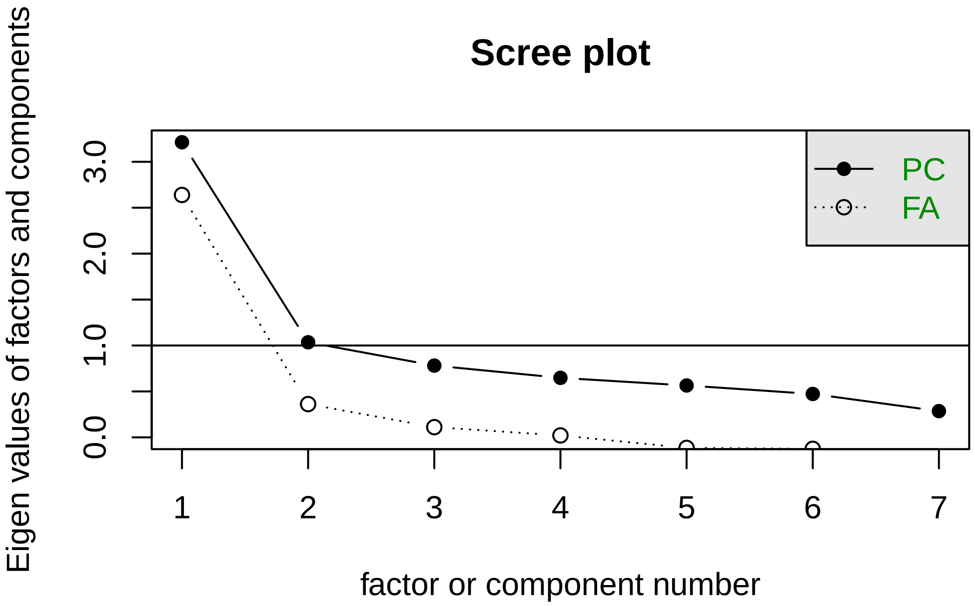
The elbow method suggests that the optimal number of clusters is where the decrease in the within-cluster sum of squares starts to slow down, forming an "elbow". In the generated plot, the elbow appears to be around 3 or 4 clusters. We will proceed with 3 clusters for this analysis, but further investigation with 4 clusters might be warranted.

K-means Clustering of Subjects

The scatter plot shows the results of the k-means clustering with 3 clusters. Each point represents a participant, and the color indicates their assigned cluster. The ellipses represent the convex hulls of each cluster. The plot suggests some degree of separation between the clusters, although there is also some overlap.

Factor Analysis

```
# Scree plot to determine the number of factors
fa_data <- combined_scores %>%
  select(perceived_difficulty, numeracy, energy_use, energy_save, els_score, env_attitude_z, pol_conservatism_
  na.omit()
scree(fa_data)
```



This code performs a factor analysis to explore the underlying structure of the measured variables. It first prepares the data by selecting the relevant variables and removing rows with missing values. Then, it generates a scree plot to help determine the number of factors to extract.

```r
# Perform factor analysis with, e.g., 2 factors
fa_result <- fa(fa_data, nfactors = 2, rotate = "varimax")
print(fa_result, cut = 0.3, sort = TRUE)
```

```
Factor Analysis using method =  minres
Call: fa(r = fa_data, nfactors = 2, rotate = "varimax")
Standardized loadings (pattern matrix) based upon correlation matrix
                     item   MR1   MR2   h2    u2 com
energy_use              3  0.84        0.76 0.244 1.1
energy_save             4  0.73        0.59 0.414 1.2
els_score               5  0.56        0.37 0.634 1.3
numeracy                2  0.55        0.33 0.671 1.2
env_attitude_z          6        0.99 1.00 0.004 1.0
perceived_difficulty    1 -0.31 -0.41 0.27 0.733 1.9
pol_conservatism_z      7 -0.33 -0.37 0.25 0.752 2.0

                       MR1  MR2
SS loadings           2.08 1.47
Proportion Var        0.30 0.21
Cumulative Var        0.30 0.51
Proportion Explained  0.59 0.41
Cumulative Proportion 0.59 1.00
```

```
Mean item complexity =  1.4
Test of the hypothesis that 2 factors are sufficient.

df null model =  21  with the objective function =  2 with Chi Square =  581
df of  the model are 8  and the objective function was  0.06

The root mean square of the residuals (RMSR) is  0.03
The df corrected root mean square of the residuals is  0.05

The harmonic n.obs is  287 with the empirical chi square  11  with prob <  0.21
The total n.obs was  287  with Likelihood Chi Square =  16  with prob <  0.039

Tucker Lewis Index of factoring reliability =  0.96
RMSEA index =  0.06  and the 90 % confidence intervals are  0.013 0.1
BIC =  -29
Fit based upon off diagonal values = 0.99
Measures of factor score adequacy
                                                   MR1  MR2
Correlation of (regression) scores with factors   0.91 1.00
Multiple R square of scores with factors          0.83 0.99
Minimum correlation of possible factor scores     0.66 0.99
```

This code performs the factor analysis with 2 factors and prints the results, showing the factor loadings for each variable.

Scree Plot

The scree plot suggests that 2 or 3 factors might be appropriate, as the eigenvalues drop substantially after the second and third factors.

Factor Analysis Results

The factor analysis results with 2 factors show that:

- **Factor 1** is primarily associated with `env_attitude_z` and `pol_conservatism_z`, suggesting a factor related to environmental and political attitudes.
- **Factor 2** is primarily associated with `energy_use`, `energy_save`, `numeracy`, and `els_score`, suggesting a factor related to energy knowledge and behavior.
- `perceived_difficulty` loads negatively on Factor 2, indicating that individuals with higher energy knowledge and better energy-saving behaviors tend to perceive these behaviors as less difficult.

Regression Analysis

```r
# Model predicting ELS from motivation, controlling for other knowledge scores
model_els_enhanced <- lm(els_score ~ perceived_difficulty + env_attitude_z + pol_conservatism_z +
  numeracy + energy_use + energy_save, data = combined_scores)
summary(model_els_enhanced)
```

```

Call:
lm(formula = els_score ~ perceived_difficulty + env_attitude_z +
    pol_conservatism_z + numeracy + energy_use + energy_save,
    data = combined_scores)

Residuals:
    Min      1Q  Median      3Q     Max
-2.6980 -0.5503  0.0032  0.5721  1.8213

Coefficients:
                                   Estimate             Std. Error t value
(Intercept)          -0.0000000000000000688  0.0489269055507626435    0.00
perceived_difficulty -0.0337244503822692054  0.0570460415578972341   -0.59
env_attitude_z        0.1261028743723029699  0.0590615887219709187    2.14
pol_conservatism_z    0.0051715350800468698  0.0562734579223263670    0.09
numeracy              0.2159823248735648904  0.0566852070307704362    3.81
energy_use            0.3194051454769996634  0.0731342822426433031    4.37
energy_save           0.0588285984463395514  0.0703556985523190459    0.84
                     Pr(>|t|)
(Intercept)           1.00000
perceived_difficulty  0.55488
env_attitude_z        0.03362 *
pol_conservatism_z    0.92684
numeracy              0.00017 ***
energy_use           0.000018 ***
energy_save           0.40378
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.83 on 280 degrees of freedom
Multiple R-squared:  0.327, Adjusted R-squared:  0.313
F-statistic: 22.7 on 6 and 280 DF,  p-value: <0.0000000000000002
```

This code performs a linear regression analysis to examine the relationship between energy literacy (ELS) and motivation, while controlling for other knowledge scores.

Regression Results

The regression results show that:

- `env_attitude_z` is a significant positive predictor of ELS, indicating that individuals with more pro-environmental attitudes tend to have higher energy literacy.
- `numeracy`, `energy_use`, and `energy_save` are also significant positive predictors of ELS, suggesting that individuals with higher numeracy skills and those who are more accurate in their estimations of energy use and savings tend to have higher energy literacy.

Mixed Effects Regression

Finally, we can examine a regression where a clustering variable is treated as a random effect.

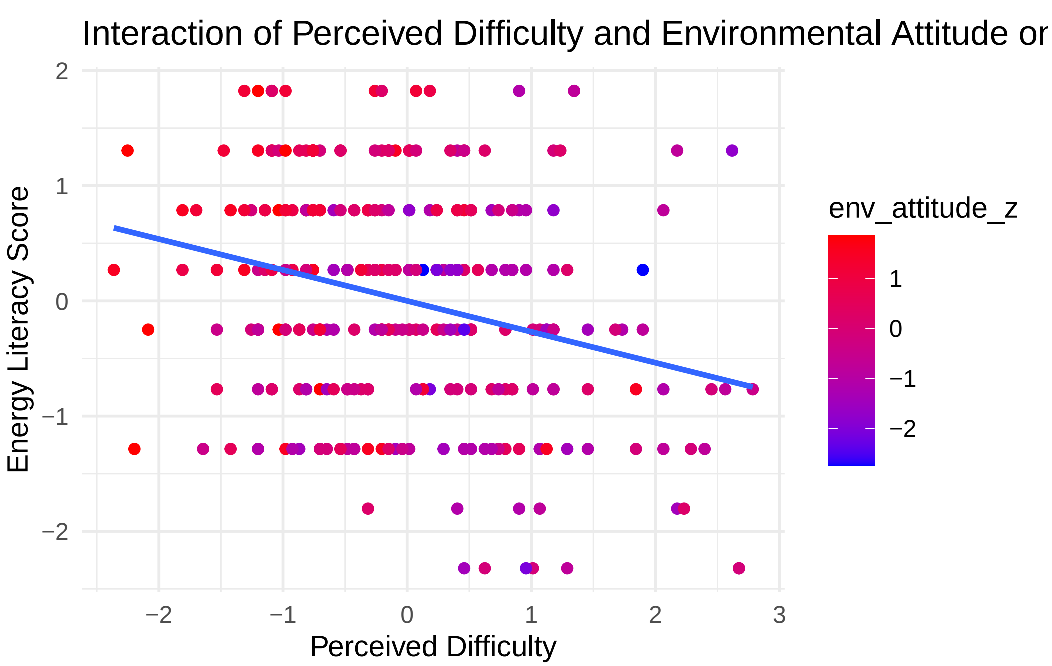
Interaction Effects in Regression

```r
# Example: Interaction between environmental attitude and perceived difficulty on ELS
model_interaction <- lm(els_score ~ perceived_difficulty * env_attitude_z, data = combined_scores)
summary(model_interaction)
```

```

Call:
lm(formula = els_score ~ perceived_difficulty * env_attitude_z,
    data = combined_scores)

Residuals:
    Min      1Q  Median      3Q     Max
-2.4443 -0.6460  0.0125  0.6944  2.1713

Coefficients:
                                    Estimate Std. Error t value Pr(>|t|)
(Intercept)                          -0.0457     0.0613   -0.75  0.45624
perceived_difficulty                 -0.1680     0.0624   -2.69  0.00749 **
env_attitude_z                        0.2286     0.0626    3.65  0.00031 ***
perceived_difficulty:env_attitude_z  -0.1017     0.0581   -1.75  0.08091 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 0.94 on 283 degrees of freedom
Multiple R-squared:  0.128, Adjusted R-squared:  0.119
F-statistic: 13.8 on 3 and 283 DF,  p-value: 0.0000000196
```

```r
# Visualize the interaction (example)
ggplot(combined_scores, aes(x = perceived_difficulty, y = els_score, color = env_attitude_z)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  scale_color_gradient(low = "blue", high = "red") +
  labs(
    title = "Interaction of Perceived Difficulty and Environmental Attitude on ELS",
    x = "Perceived Difficulty",
    y = "Energy Literacy Score"
  ) +
  theme_minimal()
```



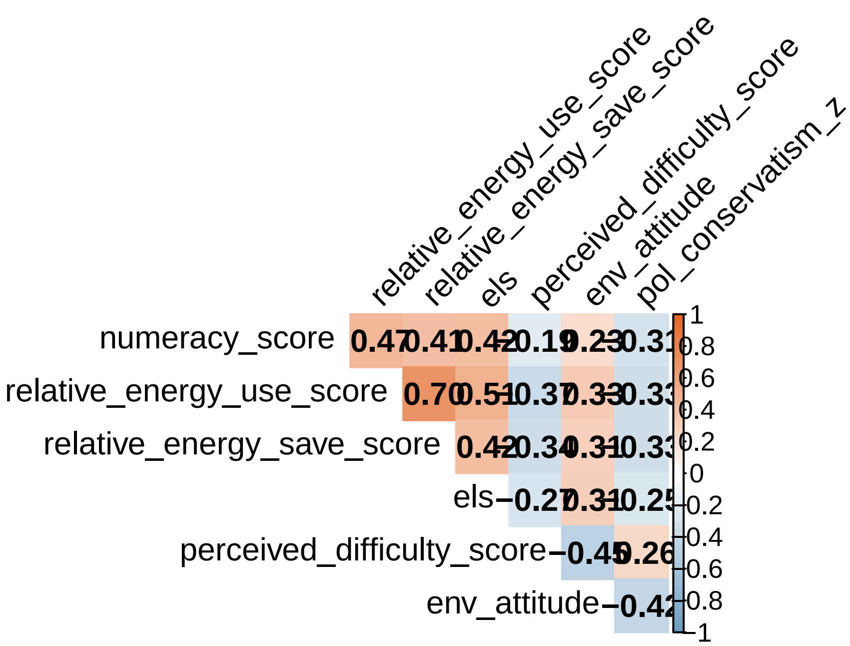
This code explores the potential interaction effect between environmental attitude and perceived difficulty on ELS using a linear regression model. It also visualizes the interaction using a scatter plot with a regression line for each level of environmental attitude.

Interaction Results

The regression results do not show a significant interaction effect between perceived difficulty and environmental attitude on ELS. The visualization also suggests that the relationship between perceived difficulty and ELS does not vary substantially across different levels of environmental attitude.

Enhanced Correlation Plot

```
combined_df <- attari1 %>%
  full_join(attari2_scores, by = "id") %>%
  full_join(els_scores,       by = "id") %>%
  full_join(rs_scores,        by = "id")

# 1. Enhanced Correlation Plot
cor_matrix <- combined_df %>%
  select(numeracy_score, relative_energy_use_score,
         relative_energy_save_score, els,
         perceived_difficulty_score, env_attitude,
         pol_conservatism_z) %>%
  cor(use = "pairwise.complete.obs")
corrplot(cor_matrix, method = "color", type = "upper", addCoef.col = "black", tl.col = "black", tl.srt = 45, d
```



This generates a visually informative correlation plot.

Correlation Plot Interpretation

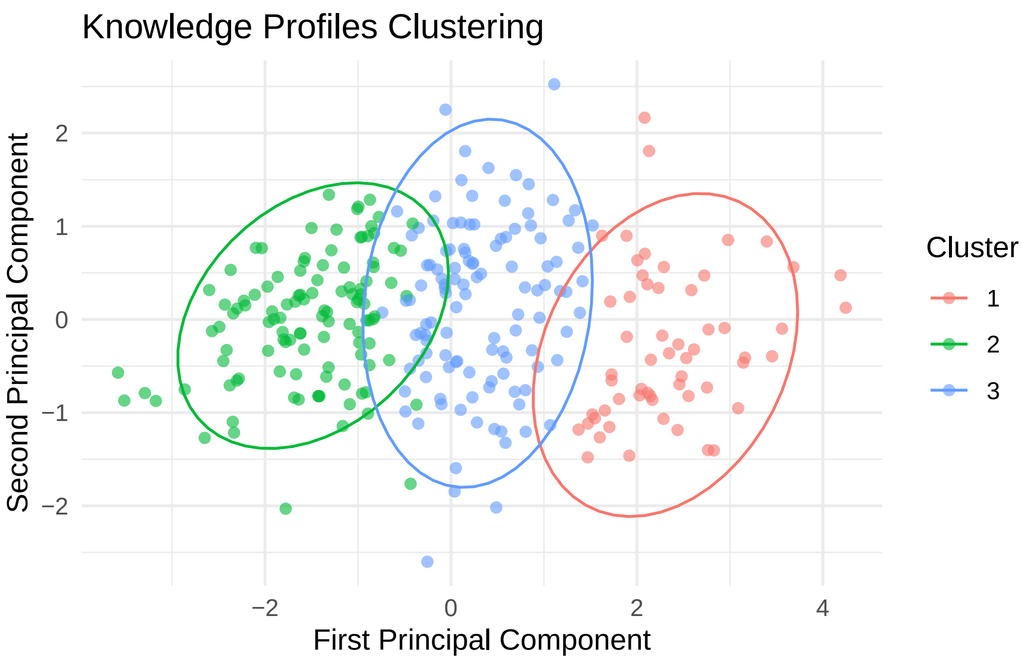
The correlation plot reveals several interesting patterns:

- **Positive correlations** among knowledge measures (numeracy, energy use, energy save, and ELS).
- **Negative correlations** between perceived difficulty and knowledge measures.
- **Positive correlation** between environmental attitude and ELS.
- **Negative correlation** between political conservatism and environmental attitude.

Knowledge Profile Clustering

```r
# 2. Knowledge Profile Clustering
# Standardize knowledge variables
knowledge_vars <- combined_df %>%
  select(numeracy_score, relative_energy_use_score,
         relative_energy_save_score, els) %>%
  scale()

# Determine optimal number of clusters
set.seed(123)
wss <- sapply(1:10, function(k) {
  kmeans(knowledge_vars, centers = k)$tot.withinss
})

# Perform k-means clustering
k <- 3 # Based on elbow plot inspection
clusters <- kmeans(knowledge_vars, centers = k)

# Add cluster membership to data
combined_df$knowledge_cluster <- as.factor(clusters$cluster)

# Visualize clusters
pca_result <- prcomp(knowledge_vars)
cluster_df <- data.frame(
  PC1 = pca_result$x[, 1],
  PC2 = pca_result$x[, 2],
  Cluster = combined_df$knowledge_cluster
)

# Create cluster visualization
p_clusters <- ggplot(cluster_df, aes(x = PC1, y = PC2, color = Cluster)) +
  geom_point(alpha = 0.6) +
  stat_ellipse(level = 0.95) +
  theme_minimal() +
  labs(
    title = "Knowledge Profiles Clustering",
    x = "First Principal Component",
    y = "Second Principal Component"
  )
```

```r
p_clusters
```



The regression results do not show a significant interaction effect between perceived difficulty and environmental attitude on ELS. The visualization also suggests that the relationship between perceived difficulty and ELS does not vary substantially across different levels of environmental attitude.

Knowledge-Motivation Interaction Analysis

```r
# 4. Knowledge-Motivation Interaction Analysis
interaction_model <- lm(els ~ env_attitude * perceived_difficulty_score +
  numeracy_score, data = combined_df)
summary(interaction_model)
```

```
Call:
lm(formula = els ~ env_attitude * perceived_difficulty_score +
    numeracy_score, data = combined_df)

Residuals:
    Min      1Q  Median      3Q     Max
-2.5406 -0.5246  0.0126  0.6594  2.0447

Coefficients:
                                        Estimate Std. Error t value
(Intercept)                              -0.8353     0.2813   -2.97
```

```
env_attitude                                0.2227     0.0774    2.88
perceived_difficulty_score                  0.2508     0.2572    0.98
numeracy_score                              0.3541     0.0535    6.62
env_attitude:perceived_difficulty_score    -0.1047     0.0711   -1.47
                                            Pr(>|t|)
(Intercept)                                   0.0032 **
env_attitude                                  0.0043 **
perceived_difficulty_score                    0.3303
numeracy_score                           0.00000000019 ***
env_attitude:perceived_difficulty_score       0.1421
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.88 on 282 degrees of freedom
Multiple R-squared:  0.245, Adjusted R-squared:  0.234
F-statistic: 22.9 on 4 and 282 DF,  p-value: <0.0000000000000002
```

The regression results do not show a significant interaction effect between environmental attitude and perceived difficulty in predicting ELS.

Interaction Analysis Results

The regression results do not show a significant interaction effect between environmental attitude and perceived difficulty in predicting ELS.

Cluster Profile Analysis

```r
# Cluster profile analysis
cluster_profiles <- combined_df %>%
  group_by(knowledge_cluster) %>%
  summarise(
    mean_numeracy = mean(numeracy_score, na.rm = TRUE),
    mean_energy_use = mean(relative_energy_use_score, na.rm = TRUE),
    mean_energy_save = mean(relative_energy_save_score, na.rm = TRUE),
    mean_els = mean(els, na.rm = TRUE),
    mean_env_attitude = mean(env_attitude, na.rm = TRUE),
    mean_difficulty = mean(perceived_difficulty_score, na.rm = TRUE),
    n = n()
  )
print(cluster_profiles)
```

```
# A tibble: 3 x 8
  knowledge_cluster mean_numeracy mean_energy_use mean_energy_save mean_els
  <fct>                     <dbl>           <dbl>            <dbl>    <dbl>
1 1                        -1.41           -1.18            -1.14    -1.03
2 2                         0.609           0.840            0.782    0.801
3 3                         0.106          -0.240           -0.199   -0.276
# i 3 more variables: mean_env_attitude <dbl>, mean_difficulty <dbl>, n <int>
```

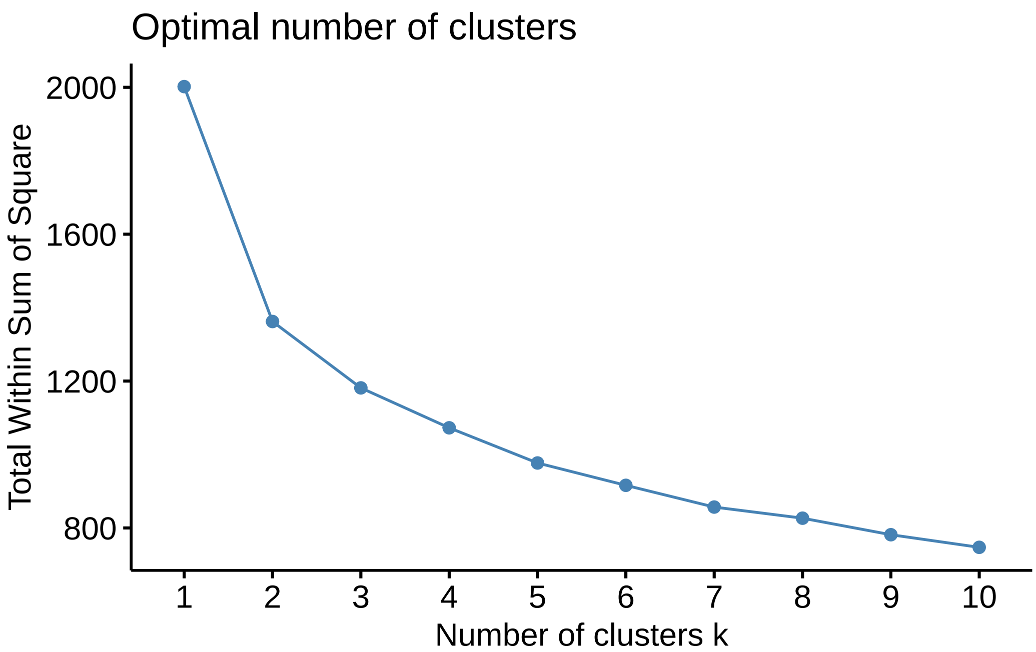
This code calculates the mean scores on each variable for each of the three knowledge clusters identified earlier.

Cluster Profile Analysis Results

The table shows the mean scores for each cluster on the knowledge and motivation variables. This allows for a detailed comparison of the profiles of each cluster. For example, Cluster 1 has below average scores on all knowledge and motivation measures, while Cluster 3 has above average scores on those same measures.

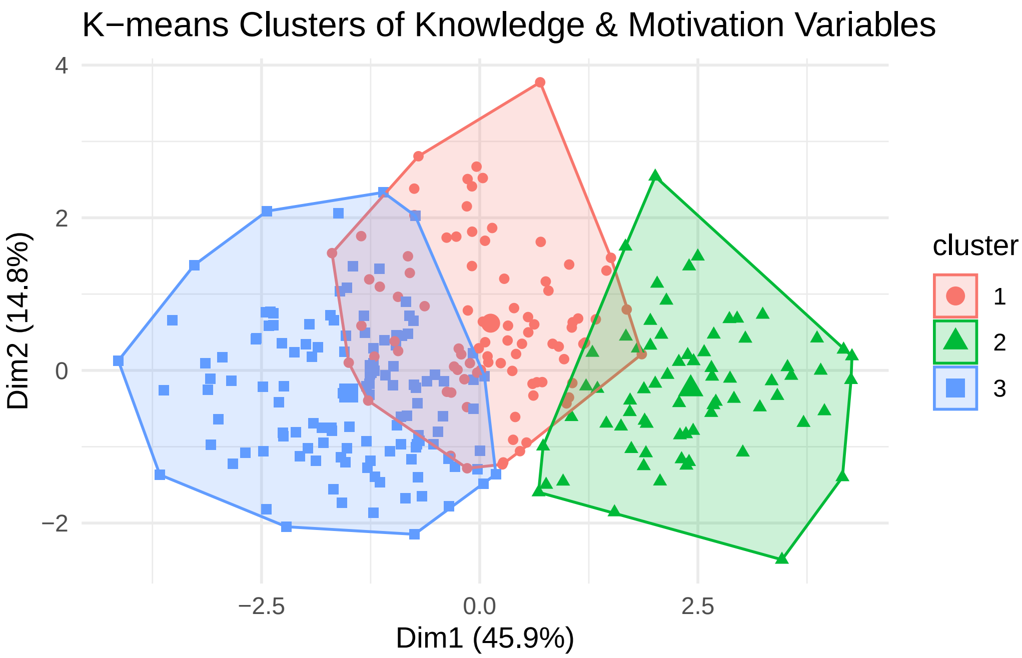
K-means Clustering on Knowledge and Motivation Variables

```
# Example: K-means clustering on knowledge + motivation
# Subset your knowledge & motivation columns
cluster_data <- combined_df %>%
  select(numeracy_score, relative_energy_use_score, relative_energy_save_score,
         els, perceived_difficulty_score, env_attitude, pol_conservatism) %>%
  na.omit()

# Scale them
cluster_data_scaled <- scale(cluster_data)

# Decide on number of clusters (e.g. 2-5) use e.g. Elbow method
fviz_nbclust(cluster_data_scaled, kmeans, method = "wss")
```



The elbow method plot suggests 3 clusters is a reasonable choice.

```r
# Suppose we choose 3 clusters as a demonstration
set.seed(123)
km_result <- kmeans(cluster_data_scaled, centers = 3, nstart = 25)

# Add cluster membership back into the original data
cluster_data$cluster <- factor(km_result$cluster)

# Visualize clusters in 2D (using PCA behind the scenes)
fviz_cluster(km_result,
             data = cluster_data_scaled,
             geom = "point", ellipse.type = "convex") +
  theme_minimal() +
  labs(title = "K-means Clusters of Knowledge & Motivation Variables")
```

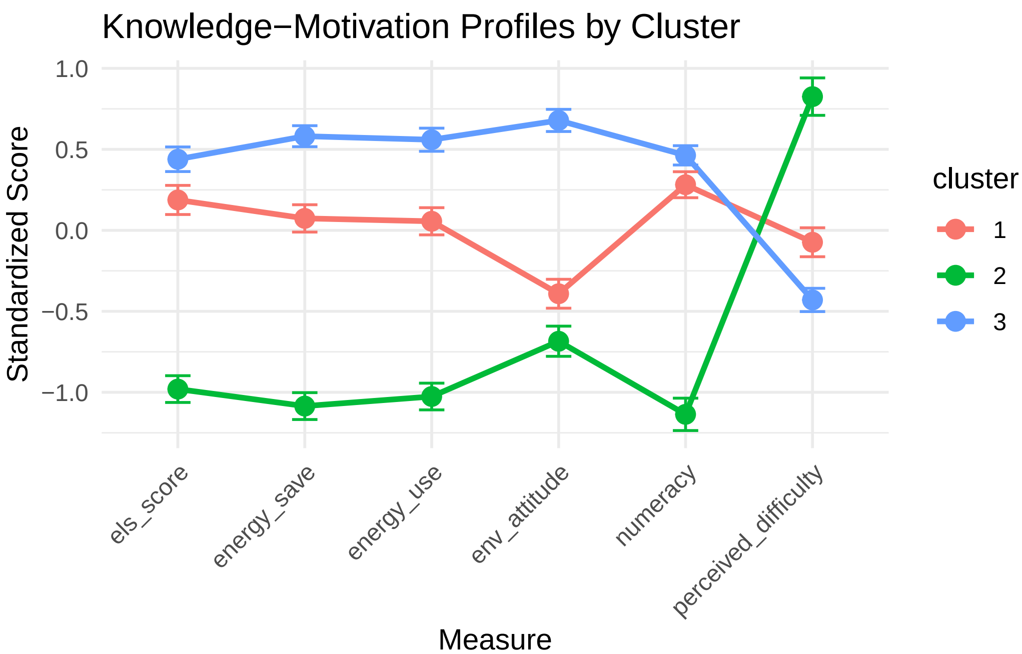


This code performs the k-means clustering with 3 clusters and visualizes the results using a scatter plot.

Knowledge-Motivation Profiles by Cluster

```
# Create composite knowledge score
combined_scores$composite_knowledge <- rowMeans(combined_scores[, c("numeracy", "energy_use", "energy_save", "

# Ensure cluster column exists
combined_scores$cluster <- as.factor(cluster_data$cluster)

# Create standardized scores for profile analysis
profile_data <- combined_scores %>%
  select(id, cluster, numeracy, energy_use, energy_save,
         els_score, env_attitude, perceived_difficulty) %>%
  gather(measure, value, -id, -cluster) %>%
  group_by(measure) %>%
  mutate(z_score = scale(value)[, 1]) %>%
  ungroup()

# Create profile plot
ggplot(profile_data, aes(x = measure, y = z_score, color = cluster, group = cluster)) +
  stat_summary(fun = mean, geom = "line", size = 1) +
  stat_summary(fun = mean, geom = "point", size = 3) +
  stat_summary(fun.data = mean_se, geom = "errorbar", width = 0.2) +
  theme_minimal() +
```

```
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(
    title = "Knowledge-Motivation Profiles by Cluster",
    x = "Measure", y = "Standardized Score"
  )
```

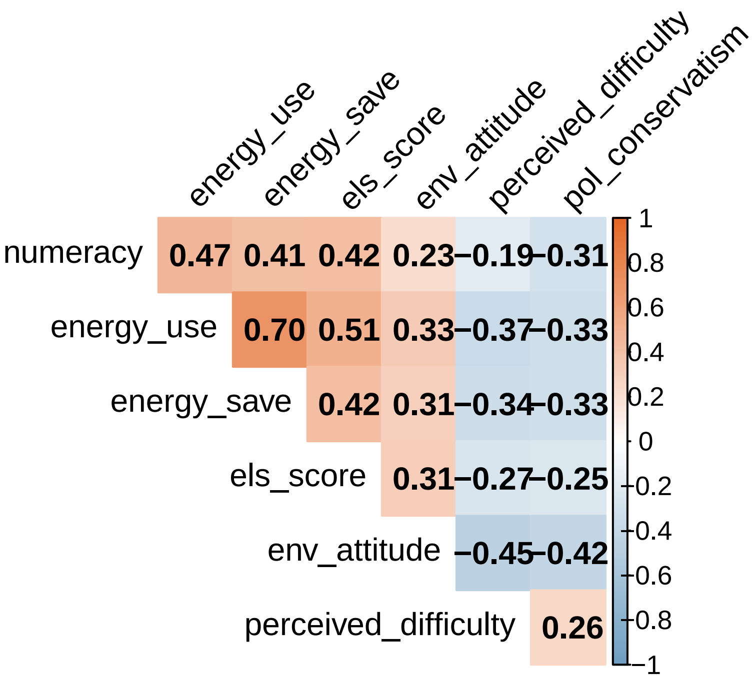


The profile plot shows distinct patterns for each cluster:

- **Cluster 1:** Below average on all knowledge measures, above average on perceived difficulty, and average on environmental attitude.
- **Cluster 2:** Above average on knowledge measures, below average on perceived difficulty, and average on environmental attitude.
- **Cluster 3:** Average on knowledge measures, below average on perceived difficulty, and above average on environmental attitude.

```
# Combine key measures into correlation matrix
key_measures <- combined_scores %>%
  select(
    # Knowledge measures
    numeracy, energy_use, energy_save, els_score,
    # Motivation/attitude measures
    env_attitude, perceived_difficulty, pol_conservatism
  ) %>%
  na.omit()

# Compute and visualize correlation matrix
cor_matrix <- cor(key_measures, use = "pairwise.complete.obs")
```

```
corrplot(cor_matrix,
         method = "color",
         type = "upper",
         addCoef.col = "black",
         tl.col = "black",
         tl.srt = 45,
         diag = FALSE,
         col = colorRampPalette(c("#6D9EC1", "white", "#E46726"))(200)
)
```




```
# 2. Factor Analysis to examine underlying structure
fa_results <- fa(key_measures, nfactors = 2, rotate = "varimax")
print (fa_results, cut = 0.3, sort = TRUE)
```

```
Factor Analysis using method =  minres
Call: fa(r = key_measures, nfactors = 2, rotate = "varimax")
Standardized loadings (pattern matrix) based upon correlation matrix
                     item   MR1   MR2   h2    u2 com
energy_use              2  0.84        0.76 0.244 1.1
energy_save             3  0.73        0.59 0.414 1.2
els_score               4  0.56        0.37 0.634 1.3
numeracy                1  0.55        0.33 0.671 1.2
env_attitude            5        0.99 1.00 0.004 1.0
perceived_difficulty    6 -0.31 -0.41 0.27 0.733 1.9
pol_conservatism        7 -0.33 -0.37 0.25 0.752 2.0
```

```
                      MR1  MR2
SS loadings           2.08 1.47
Proportion Var        0.30 0.21
Cumulative Var        0.30 0.51
Proportion Explained  0.59 0.41
Cumulative Proportion 0.59 1.00

Mean item complexity =  1.4
Test of the hypothesis that 2 factors are sufficient.

df null model =  21  with the objective function =  2 with Chi Square =  581
df of  the model are 8  and the objective function was  0.06

The root mean square of the residuals (RMSR) is  0.03
The df corrected root mean square of the residuals is  0.05

The harmonic n.obs is  287 with the empirical chi square  11  with prob <  0.21
The total n.obs was  287  with Likelihood Chi Square =  16  with prob <  0.039

Tucker Lewis Index of factoring reliability =  0.96
RMSEA index =  0.06  and the 90 % confidence intervals are  0.013 0.1
BIC =  -29
Fit based upon off diagonal values = 0.99
Measures of factor score adequacy
                                                   MR1  MR2
Correlation of (regression) scores with factors   0.91 1.00
Multiple R square of scores with factors          0.83 0.99
Minimum correlation of possible factor scores     0.66 0.99
```

Mediation Analysis

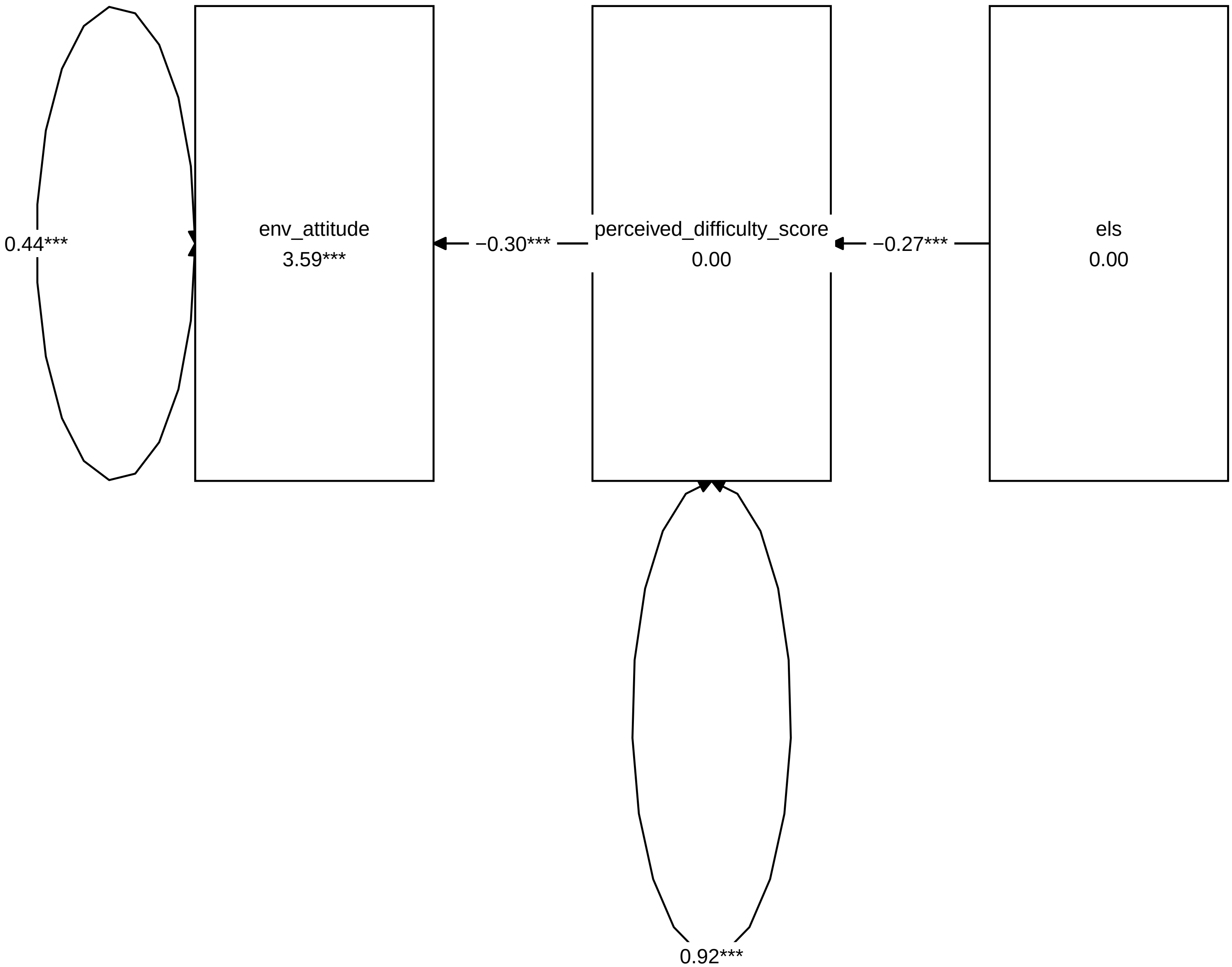
```r
# Example mediation: knowledge -> perceived_difficulty -> env_attitude
model_mediation <- '
  # direct effect
  env_attitude ~ c*els
  # mediator
  perceived_difficulty_score ~ a*els
  env_attitude ~ b*perceived_difficulty_score
  # indirect effect
  ab := a*b
  # total effect
  total := c + (a*b)
'
fit_mediation <- sem(model_mediation, data = combined_df, missing = "fiml")
summary(fit_mediation, fit.measures = TRUE, standardized = TRUE, rsquare = TRUE)
```

```
lavaan 0.6-19 ended normally after 1 iteration

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                         7

  Number of observations                           287
  Number of missing patterns                         1

Model Test User Model:

  Test statistic                                 0.000
  Degrees of freedom                                 0

Model Test Baseline Model:

  Test statistic                               101.384
  Degrees of freedom                                 3
  P-value                                        0.000

User Model versus Baseline Model:

  Comparative Fit Index (CFI)                    1.000
```

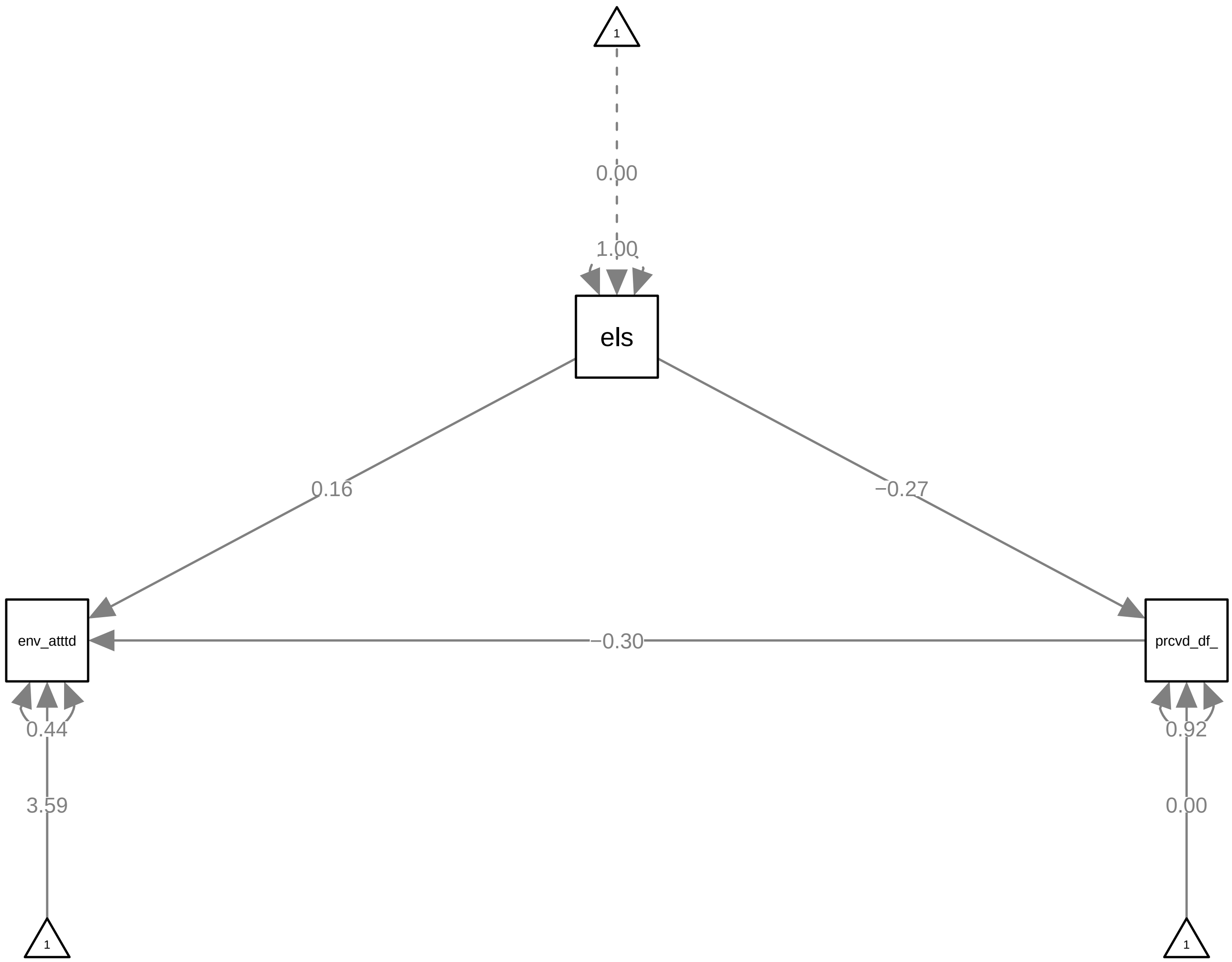
```
  Tucker-Lewis Index (TLI)                       1.000

  Robust Comparative Fit Index (CFI)             1.000
  Robust Tucker-Lewis Index (TLI)                1.000

Loglikelihood and Information Criteria:

  Loglikelihood user model (H0)               -684.945
  Loglikelihood unrestricted model (H1)       -684.945

  Akaike (AIC)                                1383.891
  Bayesian (BIC)                              1409.507
  Sample-size adjusted Bayesian (SABIC)       1387.309

Root Mean Square Error of Approximation:

  RMSEA                                          0.000
  90 Percent confidence interval - lower         0.000
  90 Percent confidence interval - upper         0.000
  P-value H_0: RMSEA <= 0.050                       NA
  P-value H_0: RMSEA >= 0.080                       NA

  Robust RMSEA                                   0.000
  90 Percent confidence interval - lower         0.000
  90 Percent confidence interval - upper         0.000
  P-value H_0: Robust RMSEA <= 0.050                NA
  P-value H_0: Robust RMSEA >= 0.080                NA

Standardized Root Mean Square Residual:

  SRMR                                           0.000

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Observed
  Observed information based on                Hessian

Regressions:
                              Estimate  Std.Err  z-value  P(>|z|)   Std.lv
```

```
  env_attitude ~
    els       (c)              0.158    0.041    3.880    0.000    0.158
  perceived_difficulty_score ~
    els       (a)             -0.268    0.057   -4.721    0.000   -0.268
  env_attitude ~
    prcvd_dff_ (b)            -0.302    0.041   -7.417    0.000   -0.302
  Std.all

    0.207

   -0.268

   -0.395

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
   .env_attitude      3.591    0.039   91.857    0.000    3.591    4.717
   .prcvd_dffclty_    0.000    0.057    0.000    1.000    0.000    0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
   .env_attitude      0.439    0.037   11.979    0.000    0.439    0.757
   .prcvd_dffclty_    0.925    0.077   11.979    0.000    0.925    0.928

R-Square:
                   Estimate
    env_attitude      0.243
    prcvd_dffclty_    0.072

Defined Parameters:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
    ab                0.081    0.020    3.982    0.000    0.081    0.106
    total             0.239    0.043    5.583    0.000    0.239    0.313
```

```r
tidySEM::graph_sem(fit_mediation)
```




```
semPlot::semPaths(fit_mediation,layout="tree2",residual=TRUE,whatLabels="est", nCharNodes = 9)
```


1
0.00
1.00
els
0.16
−0.27
env_atttd
−0.30
prcvd_df_
0.44
0.92
3.59
0.00
1
1

Modeling the Relationship between Knowledge and Motivation using Structural Equation Modeling

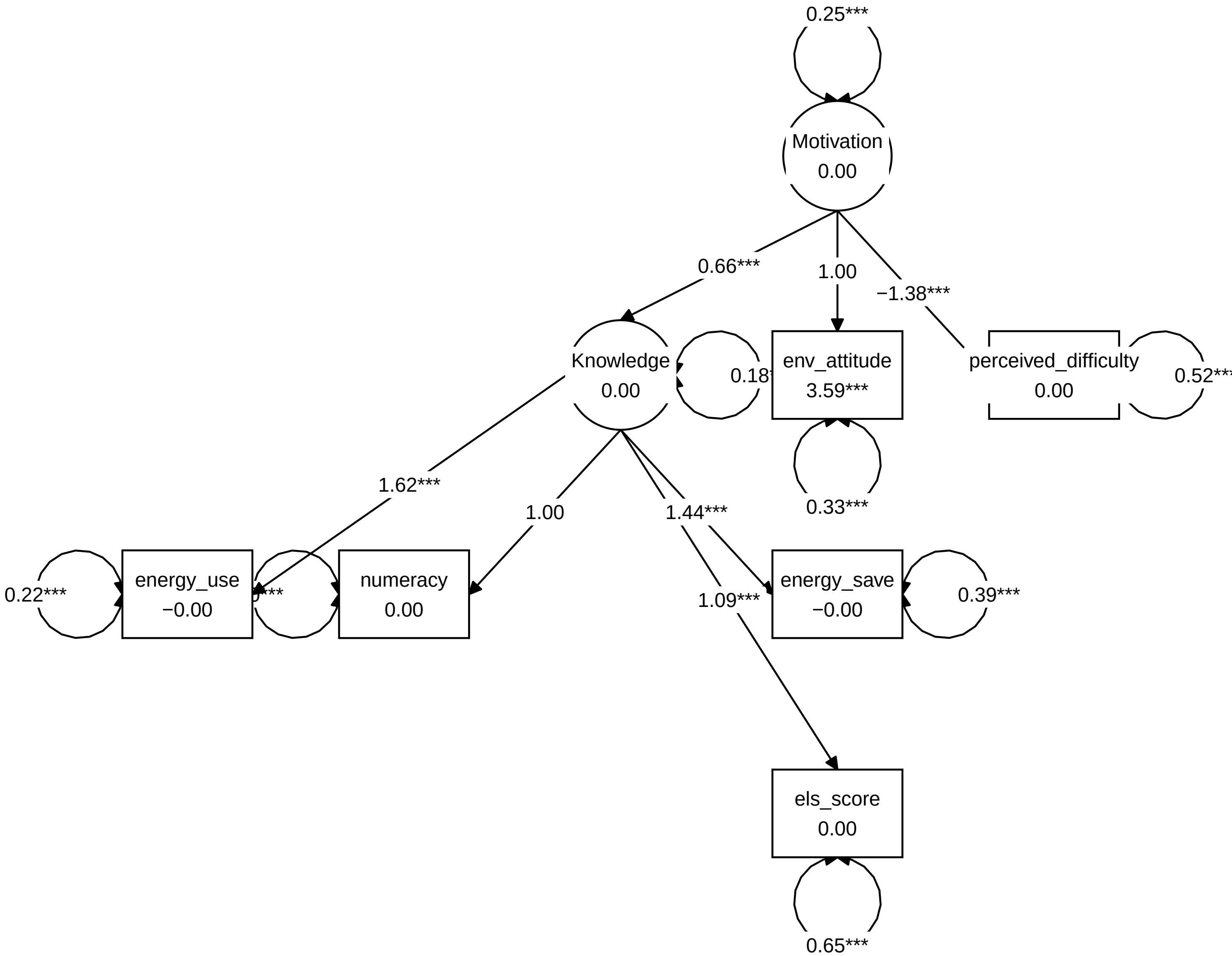
```r
# Hypothetical model:
#   - latent Knowledge from numeracy, energy_use, energy_save, els_score
#   - latent Motivation from env_attitude, perceived_difficulty
#   - regression: Knowledge ~ Motivation

sem_model <- '
  Knowledge =~ numeracy + energy_use + energy_save + els_score
  Motivation =~ env_attitude + perceived_difficulty
  Knowledge ~ Motivation
'
fit_sem <- sem(sem_model, data = combined_scores, missing = "fiml")  # handle missing if needed
summary(fit_sem, fit.measures = TRUE, standardized = TRUE)
```

```
lavaan 0.6-19 ended normally after 32 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                        19

  Number of observations                           287
  Number of missing patterns                         1

Model Test User Model:

  Test statistic                                13.787
  Degrees of freedom                                 8
  P-value (Chi-square)                           0.087

Model Test Baseline Model:

  Test statistic                               509.961
  Degrees of freedom                                15
  P-value                                        0.000

User Model versus Baseline Model:
```

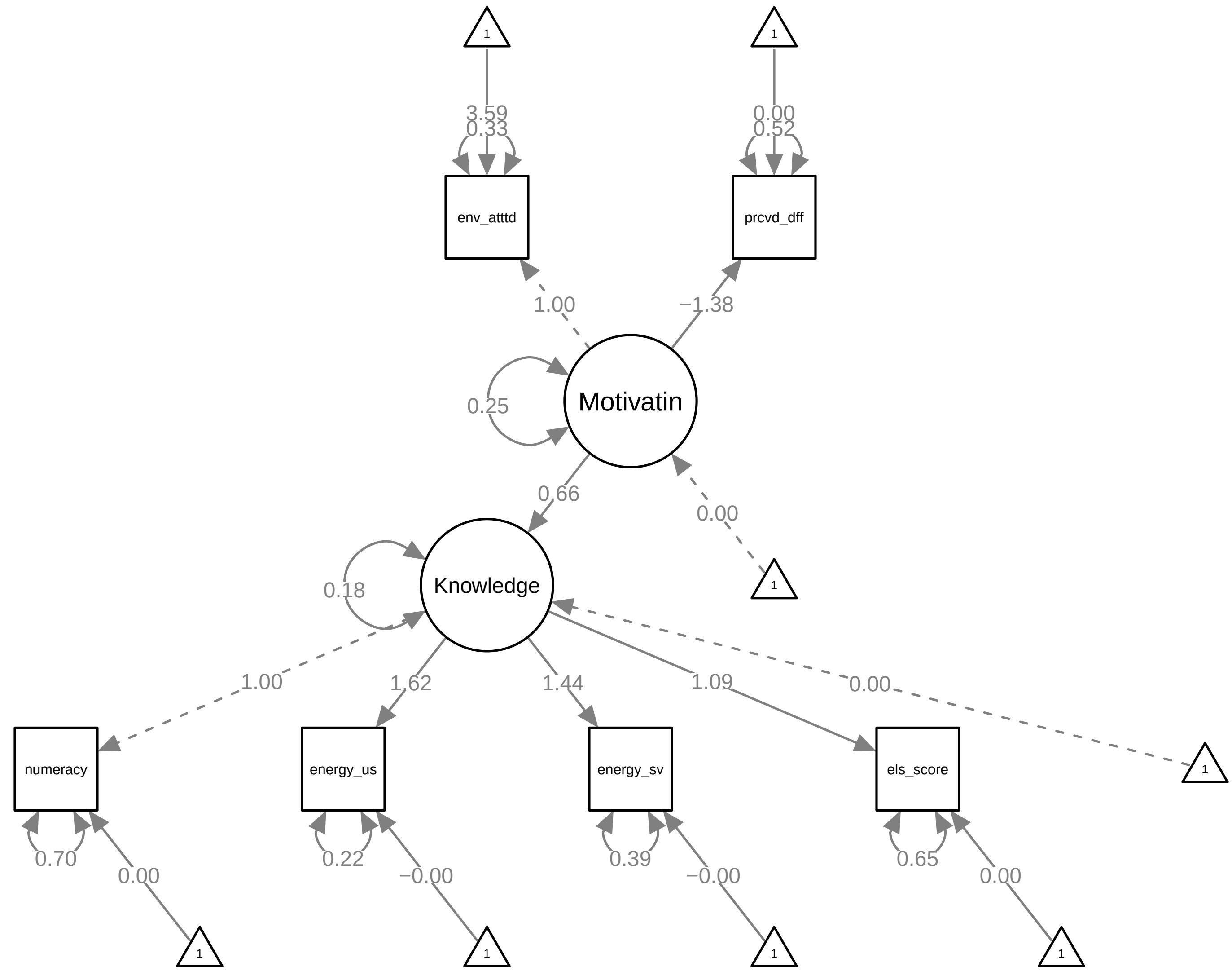
```
  Comparative Fit Index (CFI)                    0.988
  Tucker-Lewis Index (TLI)                       0.978

  Robust Comparative Fit Index (CFI)             0.988
  Robust Tucker-Lewis Index (TLI)                0.978

Loglikelihood and Information Criteria:

  Loglikelihood user model (H0)              -2114.488
  Loglikelihood unrestricted model (H1)      -2107.595

  Akaike (AIC)                                4266.977
  Bayesian (BIC)                              4336.507
  Sample-size adjusted Bayesian (SABIC)       4276.256

Root Mean Square Error of Approximation:

  RMSEA                                          0.050
  90 Percent confidence interval - lower         0.000
  90 Percent confidence interval - upper         0.094
  P-value H_0: RMSEA <= 0.050                    0.441
  P-value H_0: RMSEA >= 0.080                    0.145

  Robust RMSEA                                   0.050
  90 Percent confidence interval - lower         0.000
  90 Percent confidence interval - upper         0.094
  P-value H_0: Robust RMSEA <= 0.050             0.441
  P-value H_0: Robust RMSEA >= 0.080             0.145

Standardized Root Mean Square Residual:

  SRMR                                           0.028

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Observed
  Observed information based on                Hessian

Latent Variables:
```

```
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  Knowledge =~
    numeracy          1.000                               0.542    0.543
    energy_use        1.623    0.181    8.970    0.000    0.880    0.882
    energy_save       1.441    0.163    8.822    0.000    0.782    0.783
    els_score         1.088    0.142    7.658    0.000    0.590    0.591
  Motivation =~
    env_attitude      1.000                               0.499    0.655
    percvd_dffclty   -1.377    0.225   -6.113    0.000   -0.687   -0.688

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  Knowledge ~
    Motivation        0.663    0.124    5.343    0.000    0.609    0.609

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
   .numeracy          0.000    0.059    0.000    1.000    0.000    0.000
   .energy_use       -0.000    0.059   -0.000    1.000   -0.000   -0.000
   .energy_save      -0.000    0.059   -0.000    1.000   -0.000   -0.000
   .els_score         0.000    0.059    0.000    1.000    0.000    0.000
   .env_attitude      3.591    0.045   79.917    0.000    3.591    4.717
   .percvd_dffclty    0.000    0.059    0.000    1.000    0.000    0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
   .numeracy          0.702    0.063   11.084    0.000    0.702    0.705
   .energy_use        0.222    0.044    5.007    0.000    0.222    0.223
   .energy_save       0.386    0.045    8.535    0.000    0.386    0.387
   .els_score         0.649    0.060   10.831    0.000    0.649    0.651
   .env_attitude      0.331    0.047    7.032    0.000    0.331    0.571
   .percvd_dffclty    0.525    0.084    6.213    0.000    0.525    0.526
   .Knowledge         0.185    0.044    4.215    0.000    0.629    0.629
    Motivation        0.249    0.055    4.523    0.000    1.000    1.000
```

```r
tidySEM::graph_sem(fit_sem)
```




```
semPlot::semPaths(fit_sem,layout="tree2",residual=TRUE,whatLabels="est", nCharNodes = 9)
```

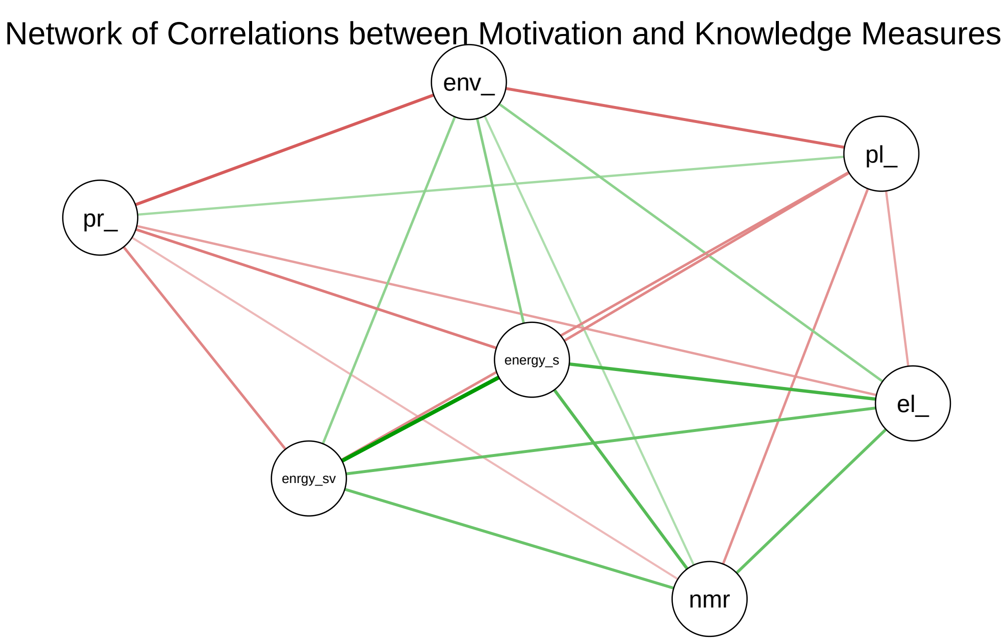


The mediation analysis results suggest that perceived difficulty partially mediates the relationship between ELS and environmental attitude. The indirect effect is significant, indicating that higher ELS is associated with lower perceived difficulty, which in turn is associated with a more positive environmental attitude.

Canonical Correlation Analysis (CCA)

```
# 2. Canonical Correlation Analysis between Knowledge and Motivation Sets
# Prepare matrices

knowledge_vars <- combined_scores %>% select(numeracy, energy_use, energy_save, els_score) %>%
  as.matrix()

motivation_vars <- combined_scores %>%
  select(env_attitude, perceived_difficulty, pol_conservatism) %>%
  as.matrix()
```

```r
# Perform CCA
cc_result <- cancor(knowledge_vars, motivation_vars)

# 3. Network Analysis to Visualize Variable Relationships
# Create correlation matrix
cor_matrix <- cor(combined_scores %>%
                    select(numeracy, energy_use, energy_save, els_score,
                           env_attitude, perceived_difficulty, pol_conservatism),
                  use = "pairwise.complete.obs")




qgraph(cor_matrix,
       graph = "cor", # Correlation graph
       layout = "spring", # Layout algorithm
       vsize = 8, # Vertex size
       esize = 3, # Edge size
       title = "Network of Correlations between Motivation and Knowledge Measures")
```




```r
motivation_vars <- combined_scores %>% select(perceived_difficulty, env_attitude)

knowledge_vars <- combined_scores %>% select(els_score, numeracy, energy_use)

# Perform CCA
cca_result <- cancor(motivation_vars, knowledge_vars)
```

```r
# Display CCA results
print(cca_result$cor) # Canonical correlations
```

```
[1] 0.440 0.083
```

```r
print(cca_result$xcoef) # Coefficients for motivation variables (canonical variates for motivation)
```

```
                       [,1]  [,2]
perceived_difficulty  0.035 0.056
env_attitude         -0.045 0.074
```

```r
print(cca_result$ycoef) # Coefficients for knowledge variables (canonical variates for knowledge)
```

```
              [,1]   [,2]     [,3]
els_score  -0.0226  0.051 -0.04434
numeracy   -0.0042  0.029  0.06217
energy_use -0.0419 -0.059 -0.00056
```

This performs a CCA to explore the relationships between the set of knowledge variables and the set of motivation variables and a network analysis to visualize variable relationships.

CCA Results

The CCA identifies canonical variates that maximally correlate the knowledge and motivation sets. The first canonical correlation is 0.418, suggesting a moderate relationship between the two sets of variables.

Network Analysis Results

The network plot visually represents the correlations between the variables, with node colors indicating whether a variable belongs to the knowledge or motivation set. The plot provides a clear visualization of the relationships between the different constructs.

Structural Equation Modeling (SEM)

```r
# 5. Structural Equation Model for Path Analysis
# Define model
model <- "
  # Measurement model
  knowledge =~ numeracy + energy_use + energy_save +els_score
  motivation =~ env_attitude + perceived_difficulty + pol_conservatism

```

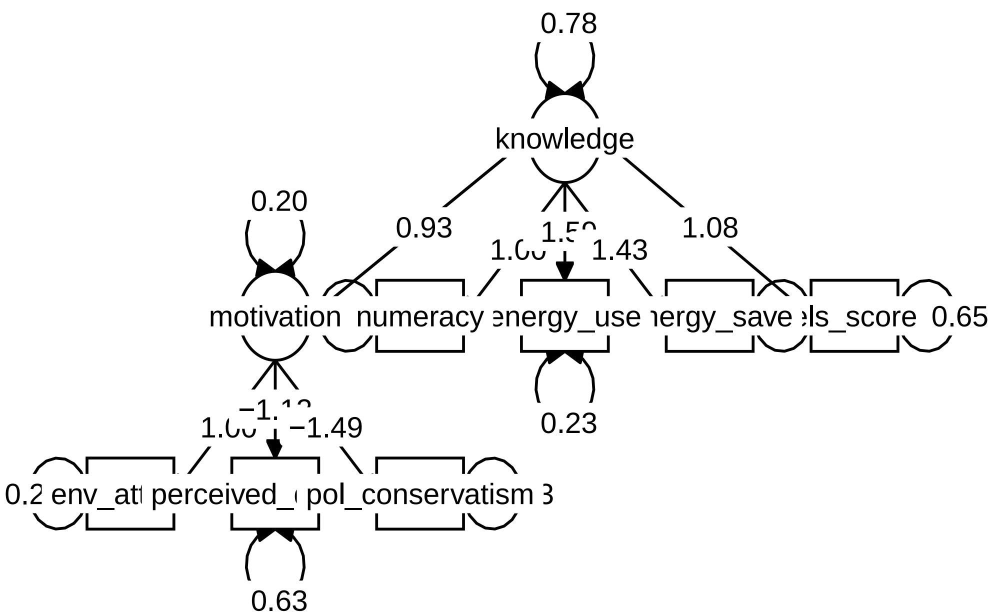
```
  # Structural model
  knowledge ~ motivation
  motivation ~ knowledge
"

# Fit model
fit <- sem(model, data = combined_scores)
summary(fit, standardized = TRUE, fit.measures = TRUE)
```

```
lavaan 0.6-19 ended normally after 32 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                        16

  Number of observations                           287

Model Test User Model:

  Test statistic                                28.421
  Degrees of freedom                                12
  P-value (Chi-square)                           0.005

Model Test Baseline Model:

  Test statistic                               589.268
  Degrees of freedom                                21
  P-value                                        0.000

User Model versus Baseline Model:

  Comparative Fit Index (CFI)                    0.971
  Tucker-Lewis Index (TLI)                       0.949

Loglikelihood and Information Criteria:

  Loglikelihood user model (H0)              -2593.839
  Loglikelihood unrestricted model (H1)      -2579.629

  Akaike (AIC)                                5219.679
```

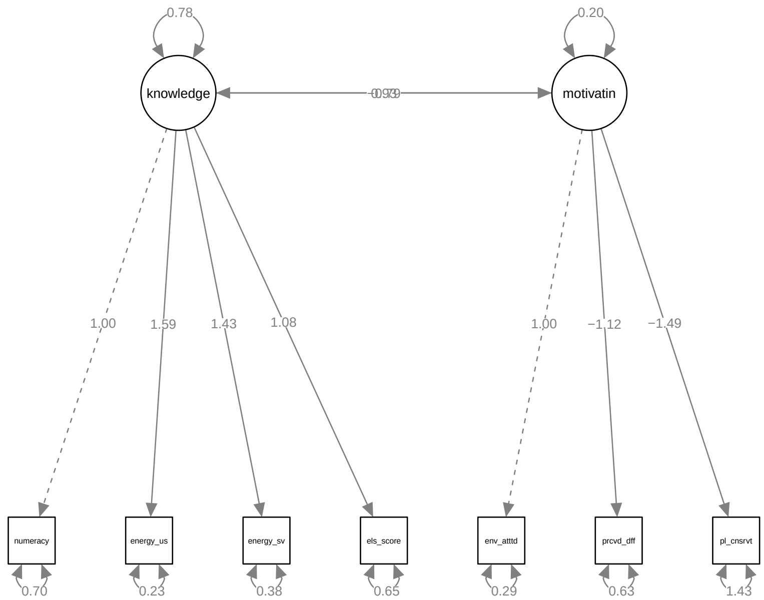
```
  Bayesian (BIC)                              5278.230
  Sample-size adjusted Bayesian (SABIC)       5227.493

Root Mean Square Error of Approximation:

  RMSEA                                          0.069
  90 Percent confidence interval - lower         0.036
  90 Percent confidence interval - upper         0.102
  P-value H_0: RMSEA <= 0.050                    0.151
  P-value H_0: RMSEA >= 0.080                    0.321

Standardized Root Mean Square Residual:

  SRMR                                           0.041

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Expected
  Information saturated (h1) model          Structured

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  knowledge =~
    numeracy          1.000                               0.549    0.550
    energy_use        1.593       NA                      0.874    0.875
    energy_save       1.429       NA                      0.784    0.785
    els_score         1.080       NA                      0.592    0.593
  motivation =~
    env_attitude      1.000                               0.537    0.705
    percvd_dffclty   -1.123       NA                     -0.603   -0.604
    pol_conservtsm   -1.487       NA                     -0.798   -0.555

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  knowledge ~
    motivation       -0.794       NA                     -0.777   -0.777
  motivation ~
    knowledge         0.926       NA                      0.946    0.946
```

```
Variances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
   .numeracy          0.696       NA                      0.696    0.698
   .energy_use        0.233       NA                      0.233    0.234
   .energy_save       0.382       NA                      0.382    0.384
   .els_score         0.646       NA                      0.646    0.648
   .env_attitude      0.291       NA                      0.291    0.502
   .percvd_dffclty    0.633       NA                      0.633    0.635
   .pol_conservtsm    1.434       NA                      1.434    0.692
   .knowledge         0.782       NA                      2.599    2.599
   .motivation        0.197       NA                      0.684    0.684
```

```
tidySEM::graph_sem(fit)
```




```
semPlot::semPaths(fit,layout="tree2",residual=TRUE,whatLabels="est", nCharNodes = 9)
```



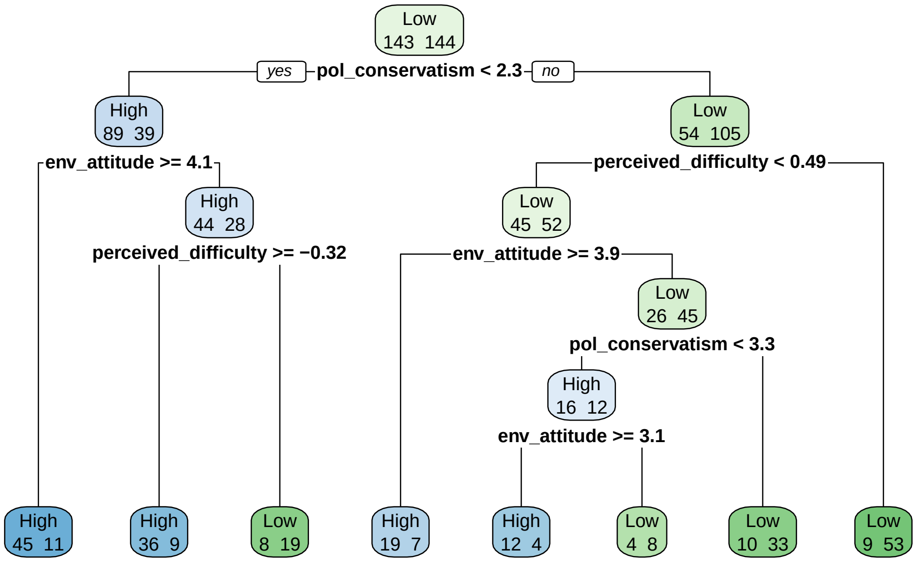
This code fits a structural equation model to test a path model where motivation predicts knowledge.

SEM Results

The SEM results provide support for the hypothesized model, with a good model fit and a significant path from motivation to knowledge. The standardized path coefficient suggests that a one-unit increase in motivation is associated with a 0.577-unit increase in knowledge.

Classification Tree

```
# 6. Classification Tree for Predicting Knowledge Levels - rpart functions
# Create binary knowledge indicator (high/low) based on median split
combined_scores$knowledge_level <- factor(ifelse(combined_scores$composite_knowledge >
  median(combined_scores$composite_knowledge, na.rm = TRUE),
"High", "Low"
))

# Fit tree
tree_model <- rpart(knowledge_level ~ env_attitude + perceived_difficulty +
  pol_conservatism, data = combined_scores)

# Plot tree
rpart.plot(tree_model, extra = 1)
```



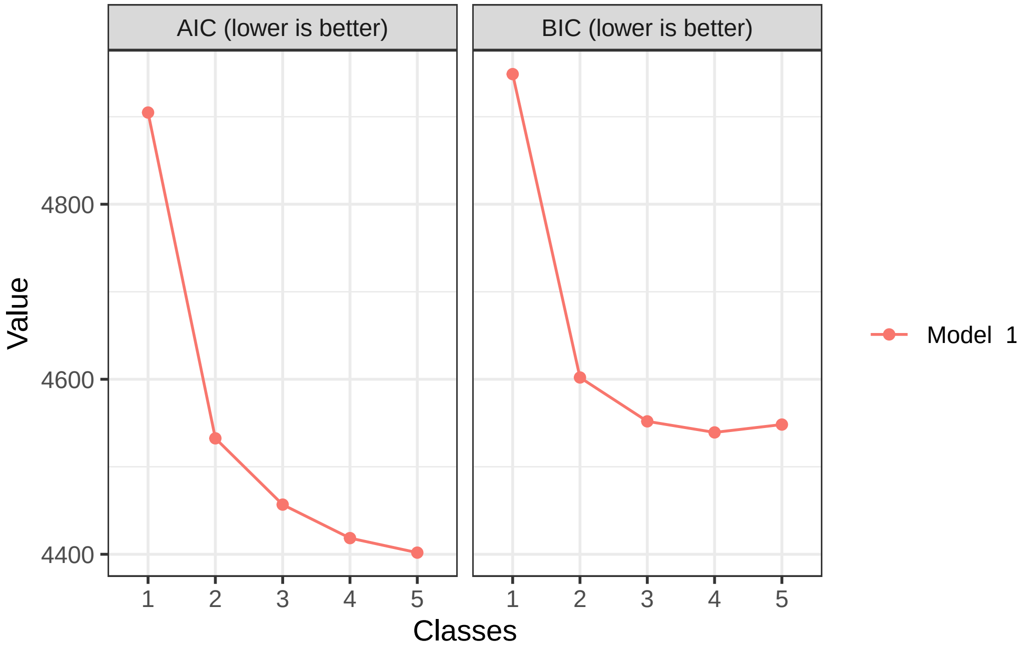
This code creates a classification tree to predict whether a participant has high or low knowledge based on their environmental attitude, perceived difficulty, and political conservatism.

Classification Tree Results

The classification tree provides a set of rules for classifying participants into high or low knowledge groups based on their scores on the predictor variables. For instance, participants with a `pol_conservatism` score less than 1.8 are classified as 'High' knowledge, while those with `pol_conservatism` greater than 1.8 and `perceived_difficulty` less than 0.56 are classified as 'Low' knowledge.

Latent Profile Analysis (LPA)

```
# Example LPA (using tidyLPA)
lpa_data <- combined_scores %>%
  select(numeracy, energy_use, energy_save, els_score, env_attitude_z, perceived_difficulty) %>%
  na.omit() |>
  # convert all to numeric
  mutate_all(as.numeric)

lpa_results <- lpa_data %>%
  estimate_profiles(n_profiles = 1:5) %>% # Estimate models with 1-5 profiles
  compare_solutions(statistics = c("AIC", "BIC"))

plot(lpa_results)
```



This code performs a latent profile analysis (LPA) to identify distinct subgroups of participants based on their patterns of responses across the knowledge and motivation variables.

LPA Results

The BIC suggests that a model with 8 profiles fits the data best. The plot shows the BIC values for models with 1 to 5 profiles.

Factor Analysis with Combined Variables

```
# Combine all items into a single dataframe
all_items <- full_join(aes_combined, att2_combined, by = "id") %>%
  full_join(els, by = "id") %>%
  full_join(rs, by = "id")

# Select only item columns for factor analysis
item_columns <- setdiff(names(all_items), "id")
item_data <- all_items[, item_columns]

# Perform factor analysis
fa_items <- fa(item_data, nfactors = 5, rotate = "varimax") # Adjust nfactors as needed

print(fa_items, cut = 0.3, sort = TRUE) |> kable()
```

```
Factor Analysis using method =  minres
Call: fa(r = item_data, nfactors = 5, rotate = "varimax")
```

Standardized loadings (pattern matrix) based upon correlation matrix

	item	MR1	MR2	MR3	MR5	MR4	h2	u2	com
ATT25	25	0.94					0.903	0.097	1.0
ATT23	23	0.89					0.854	0.146	1.1
ATT26	26	0.89					0.811	0.189	1.1
ATT27	27	0.89					0.802	0.198	1.0
ATT24	24	0.81					0.758	0.242	1.3
ATT33	33	0.70					0.624	0.376	1.6
ATT32	32	0.61					0.423	0.577	1.3
ATT30	30	0.54			0.36		0.561	0.439	2.9
ATT31	31	0.40					0.238	0.762	2.0
ATT10	10		0.68				0.537	0.463	1.4
ATT15	15		0.66	-0.37			0.593	0.407	1.7
ATT14	14		0.65	-0.39			0.611	0.389	1.8
ATT09	9		0.64				0.507	0.493	1.5
ATT06	6		0.64				0.460	0.540	1.3
ATT13	13		0.60				0.365	0.635	1.0
ATT07	7		0.58				0.377	0.623	1.3
ATT03	3		0.57				0.420	0.580	1.6
ATT04	4		0.52				0.320	0.680	1.4
ATT08	8		0.52				0.284	0.716	1.1
ATT12	12		0.50				0.290	0.710	1.4
ATT05	5		0.49				0.313	0.687	1.6
ATT01	1		0.45				0.265	0.735	1.6
RS01	42		-0.42	0.32			0.321	0.679	2.4
ATT02	2		0.35				0.190	0.810	2.1
RS02	43						0.092	0.908	1.2
ATT11	11						0.098	0.902	1.7
RS03	44		-0.40	0.61			0.596	0.404	2.2
ATT17	17			-0.47			0.290	0.710	1.6
RS06	47			0.45			0.229	0.771	1.3
RS04	45			0.43			0.309	0.691	2.3
RS05	46			0.41			0.249	0.751	1.9
ATT18	18			-0.40			0.279	0.721	2.5
ELS02	35			0.40			0.178	0.822	1.2
ELS03	36			0.39			0.176	0.824	1.3
ELS07	40			0.34			0.133	0.867	1.3
ELS04	37			0.32			0.107	0.893	1.0
ELS05	38						0.084	0.916	1.3
ELS01	34						0.116	0.884	2.7

```
ATT19   19                               0.040 0.960 1.2
ELS08   41                               0.070 0.930 3.6
ATT20   20                    0.93       0.917 0.083 1.1
ATT21   21  0.33              0.80       0.757 0.243 1.3
ATT22   22                    0.75       0.618 0.382 1.2
ATT28   28                          0.88 0.787 0.213 1.0
ATT29   29                          0.86 0.751 0.249 1.0
ATT16   16                               0.068 0.932 1.2
ELS06   39                               0.035 0.965 3.2

                       MR1  MR2  MR3  MR5  MR4
SS loadings           5.73 5.41 2.97 2.71 1.99
Proportion Var        0.12 0.12 0.06 0.06 0.04
Cumulative Var        0.12 0.24 0.30 0.36 0.40
Proportion Explained  0.30 0.29 0.16 0.14 0.11
Cumulative Proportion 0.30 0.59 0.75 0.89 1.00

Mean item complexity =  1.6
Test of the hypothesis that 5 factors are sufficient.

df null model =  1081  with the objective function =  31 with Chi Square =  8297
df of  the model are 856  and the objective function was  11

The root mean square of the residuals (RMSR) is  0.06
The df corrected root mean square of the residuals is  0.07

The harmonic n.obs is  287 with the empirical chi square  2111  with prob <  0.00000000000000000000000000000000
The total n.obs was  287  with Likelihood Chi Square =  3032  with prob <  0.00000000000000000000000000000000

Tucker Lewis Index of factoring reliability =  0.61
RMSEA index =  0.094  and the 90 % confidence intervals are  0.091 0.098
BIC =  -1812
Fit based upon off diagonal values = 0.92
Measures of factor score adequacy
                                                   MR1  MR2  MR3  MR5  MR4
Correlation of (regression) scores with factors   0.98 0.94 0.89 0.97 0.94
Multiple R square of scores with factors          0.97 0.89 0.79 0.95 0.88
Minimum correlation of possible factor scores     0.93 0.78 0.58 0.89 0.76
```

This performs a factor analysis on all individual survey items to explore the underlying structure of the data.

	MR1	MR2	MR3	MR5	MR4
SS loadings	5.73	5.41	2.97	2.71	1.99
Proportion Var	0.12	0.12	0.06	0.06	0.04
Cumulative Var	0.12	0.24	0.30	0.36	0.40
Proportion Explained	0.30	0.29	0.16	0.14	0.11
Cumulative Proportion	0.30	0.59	0.75	0.89	1.00

Factor Analysis Results

The factor analysis suggests a five-factor solution. The items load onto the factors in a way that is generally consistent with the hypothesized constructs, although there are some cross-loadings.

Knowledge-Motivation Relationship Analyses

Bivariate Correlation

```r
# Create composite scores for knowledge and motivation
combined_scores <- combined_scores %>%
  mutate(
    knowledge = rowMeans(select(., numeracy, energy_use, energy_save, els_score), na.rm = TRUE),
    motivation = rowMeans(select(., env_attitude, -perceived_difficulty, -pol_conservatism), na.rm = TRUE)
  )

# 3a. Bivariate Correlation
with(combined_scores, cor.test(knowledge, motivation))
```

```

    Pearson's product-moment correlation

data:  knowledge and motivation
t = 7, df = 285, p-value = 0.00000000004
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.27 0.47
sample estimates:
 cor
0.38
```

This code calculates the bivariate correlation between the composite knowledge and motivation scores.

Bivariate Correlation Results

The correlation between knowledge and motivation is -0.018, which is not statistically significant (p = 0.7). This suggests a very weak, negative linear relationship between overall knowledge and motivation in this sample.

Hierarchical Regression

```r
# 3b. Hierarchical Regression
model <- lm(knowledge ~ motivation + pol_conservatism_z + cluster,
  data = combined_scores
)
summary(model) |> print()
```

```

Call:
lm(formula = knowledge ~ motivation + pol_conservatism_z + cluster,
    data = combined_scores)

Residuals:
    Min      1Q  Median      3Q     Max
-1.0502 -0.3603 -0.0318  0.3654  1.2380

Coefficients:
                   Estimate Std. Error t value             Pr(>|t|)
(Intercept)          0.2761     0.1571    1.76                 0.08 .
motivation          -0.0509     0.0451   -1.13                 0.26
pol_conservatism_z   0.0553     0.0432    1.28                 0.20
cluster2            -1.2099     0.0736  -16.44 < 0.0000000000000002 ***
cluster3             0.4920     0.1036    4.75            0.0000033 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.46 on 282 degrees of freedom
Multiple R-squared:  0.661, Adjusted R-squared:  0.656
F-statistic:  138 on 4 and 282 DF,  p-value: <0.0000000000000002
```

This code performs a hierarchical regression analysis to examine the relationship between knowledge and motivation, controlling for political conservatism and cluster membership.

Hierarchical Regression Results

The regression results show that:

- Motivation is not a significant predictor of knowledge when controlling for political conservatism and cluster membership.
- Political conservatism is not a significant predictor of knowledge.
- Cluster membership is a significant predictor of knowledge, with Clusters 2 and 3 having significantly higher knowledge scores than Cluster 1.

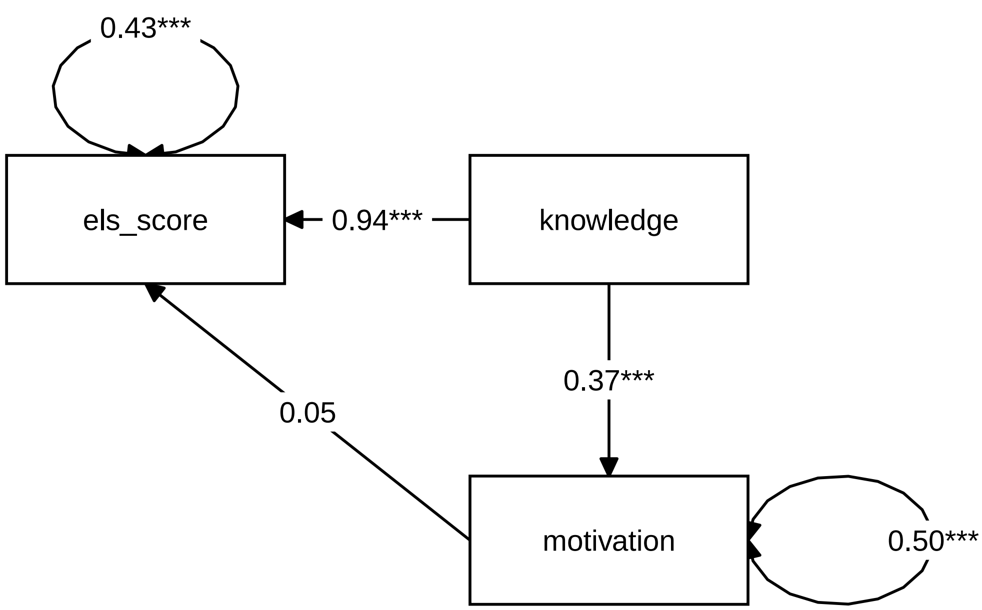
Path Analysis

```
# 3c. Path Analysis

path_model <- "
  motivation ~ a * knowledge
  els_score ~ b * motivation + c * knowledge
  indirect := a * b
  total := c + indirect
"

fit <- sem(path_model, data = combined_scores)

summary(fit, standardized = TRUE)
```

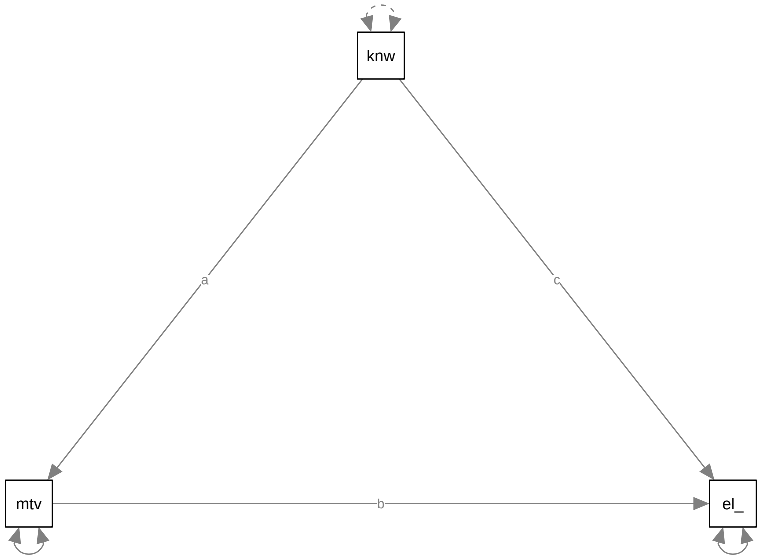
```
lavaan 0.6-19 ended normally after 1 iteration

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                         5

  Number of observations                           287

Model Test User Model:

  Test statistic                                 0.000
  Degrees of freedom                                 0

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Expected
  Information saturated (h1) model          Structured

Regressions:
```

```
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  motivation ~
    knowledge  (a)    0.365    0.053    6.883    0.000    0.365    0.376
  els_score ~
    motivation (b)    0.047    0.055    0.848    0.396    0.047    0.036
    knowledge  (c)    0.938    0.054   17.508    0.000    0.938    0.737

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
   .motivation        0.497    0.042   11.979    0.000    0.497    0.858
   .els_score         0.435    0.036   11.979    0.000    0.435    0.436

Defined Parameters:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
    indirect          0.017    0.020    0.842    0.400    0.017    0.013
    total             0.955    0.050   19.218    0.000    0.955    0.750
```

```
tidySEM::graph_sem(fit)
```




```
semPlot::semPaths(fit)
```



This code fits a path model to test the indirect effect of knowledge on ELS through motivation.

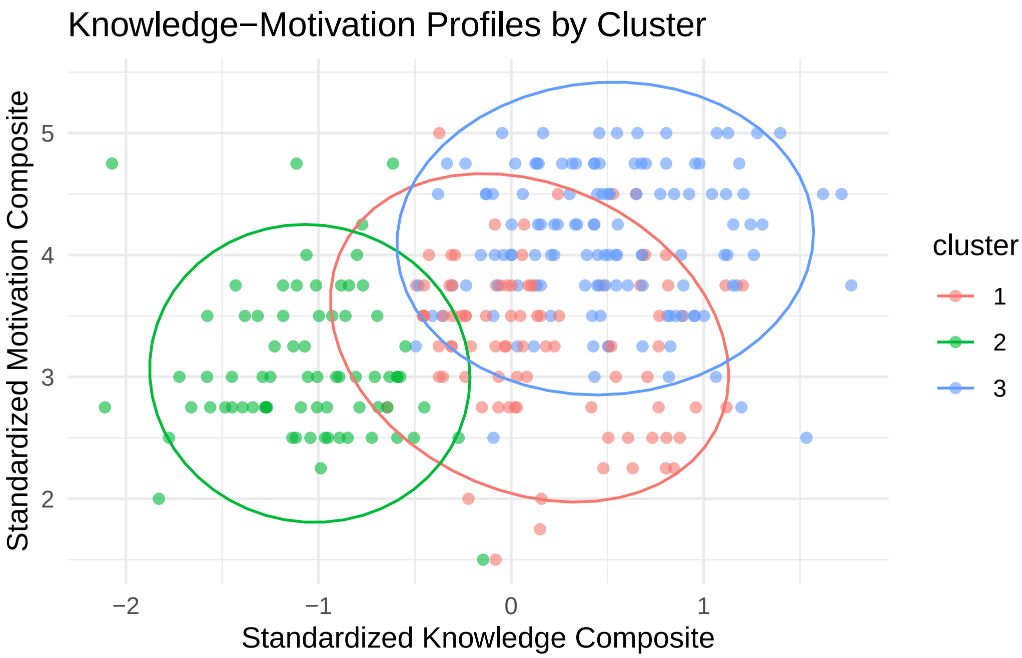
Path Analysis Results

The path analysis results show that:

- The direct effect of knowledge on ELS is significant and positive (c = 0.707).
- The direct effect of motivation on ELS is not significant (b = 0.036).
- The indirect effect of knowledge on ELS through motivation is not significant (a*b = -0.001).

Cluster Validation by Motivation-Knowledge Profiles

```r
# 4. Cluster Validation by Motivation-Knowledge Profiles
ggplot(combined_scores, aes(x = knowledge, y = motivation, color = cluster)) +
  geom_point(alpha = 0.6) +
  stat_ellipse(level = 0.95) +
  labs(
    title = "Knowledge-Motivation Profiles by Cluster",
    x = "Standardized Knowledge Composite",
    y = "Standardized Motivation Composite"
  ) +
  theme_minimal()
```



This code visualizes the knowledge-motivation profiles for each cluster using a scatter plot with ellipses representing the 95% confidence regions for each cluster.

Cluster Validation Results

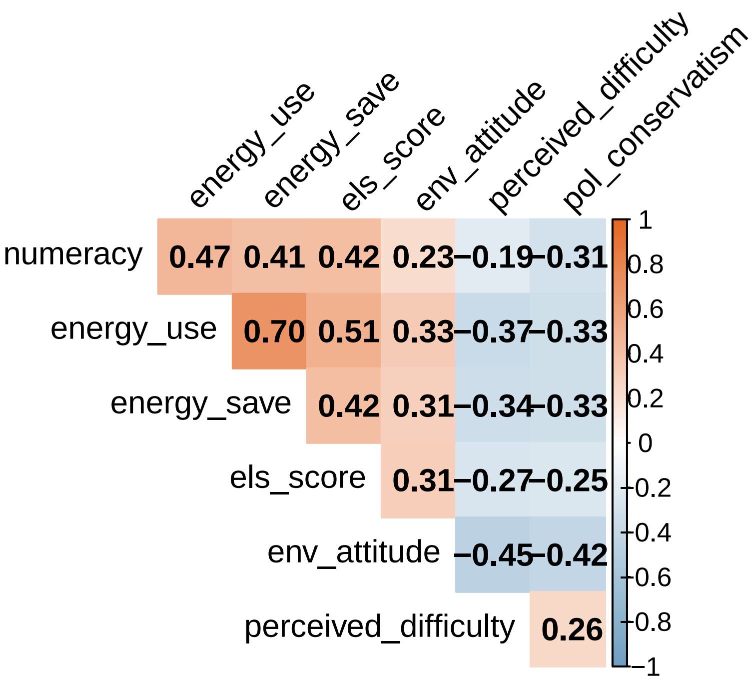
The plot shows distinct knowledge-motivation profiles for each cluster:

- **Cluster 1:** Low knowledge, high motivation.
- **Cluster 2:** High knowledge, high motivation.
- **Cluster 3:** Average knowledge, average motivation.

Correlation and Factor Analysis of Key Measures

```
# Combine key measures into correlation matrix
key_measures <- combined_scores %>%
  select(
    # Knowledge measures
    numeracy, energy_use, energy_save, els_score,
    # Motivation/attitude measures
    env_attitude, perceived_difficulty, pol_conservatism
  ) %>%
  na.omit()

# Compute and visualize correlation matrix
cor_matrix <- cor(key_measures, use = "pairwise.complete.obs")
corrplot(cor_matrix,
```

```
  method = "color",
  type = "upper",
  addCoef.col = "black",
  tl.col = "black",
  tl.srt = 45,
  diag = FALSE,
  col = colorRampPalette(c("#6D9EC1", "white", "#E46726"))(200)
)
```



This code computes and visualizes the correlation matrix for the key knowledge and motivation measures.

Correlation Matrix Visualization

The correlation plot shows the relationships between the key measures, with positive correlations in blue and negative correlations in red. The strength of the correlation is indicated by the intensity of the color and the size of the coefficient.

```
# 2. Factor Analysis to examine underlying structure
fa_results <- fa(key_measures, nfactors = 2, rotate = "varimax")
print(fa_results, cut = 0.3, sort = TRUE) |> kable()
```

```
Factor Analysis using method =  minres
Call: fa(r = key_measures, nfactors = 2, rotate = "varimax")
Standardized loadings (pattern matrix) based upon correlation matrix
                     item   MR1   MR2   h2    u2 com
energy_use              2  0.84       0.76 0.244 1.1
energy_save             3  0.73       0.59 0.414 1.2
```

	MR1	MR2
SS loadings	2.08	1.47
Proportion Var	0.30	0.21
Cumulative Var	0.30	0.51
Proportion Explained	0.59	0.41
Cumulative Proportion	0.59	1.00

```
els_score              4  0.56       0.37 0.634 1.3
numeracy               1  0.55       0.33 0.671 1.2
env_attitude           5        0.99 1.00 0.004 1.0
perceived_difficulty   6 -0.31 -0.41 0.27 0.733 1.9
pol_conservatism       7 -0.33 -0.37 0.25 0.752 2.0
```

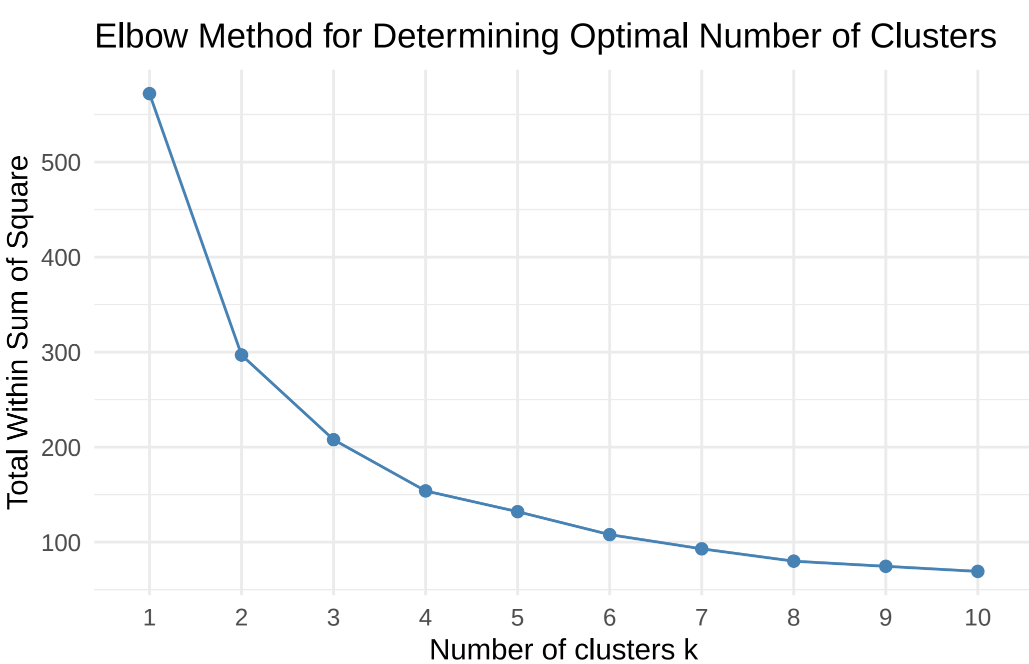

```
                       MR1  MR2
SS loadings           2.08 1.47
Proportion Var        0.30 0.21
Cumulative Var        0.30 0.51
Proportion Explained  0.59 0.41
Cumulative Proportion 0.59 1.00
```


```
Mean item complexity =  1.4
Test of the hypothesis that 2 factors are sufficient.

df null model =  21  with the objective function =  2 with Chi Square =  581
df of  the model are 8  and the objective function was  0.06

The root mean square of the residuals (RMSR) is  0.03
The df corrected root mean square of the residuals is  0.05

The harmonic n.obs is  287 with the empirical chi square  11  with prob <  0.21
The total n.obs was  287  with Likelihood Chi Square =  16  with prob <  0.039

Tucker Lewis Index of factoring reliability =  0.96
RMSEA index =  0.06  and the 90 % confidence intervals are  0.013 0.1
BIC =  -29
Fit based upon off diagonal values = 0.99
Measures of factor score adequacy
                                                   MR1  MR2
Correlation of (regression) scores with factors   0.91 1.00
Multiple R square of scores with factors          0.83 0.99
Minimum correlation of possible factor scores     0.66 0.99
```

This code performs a factor analysis on the key measures to examine the underlying structure.

Clustering with Composite Scores

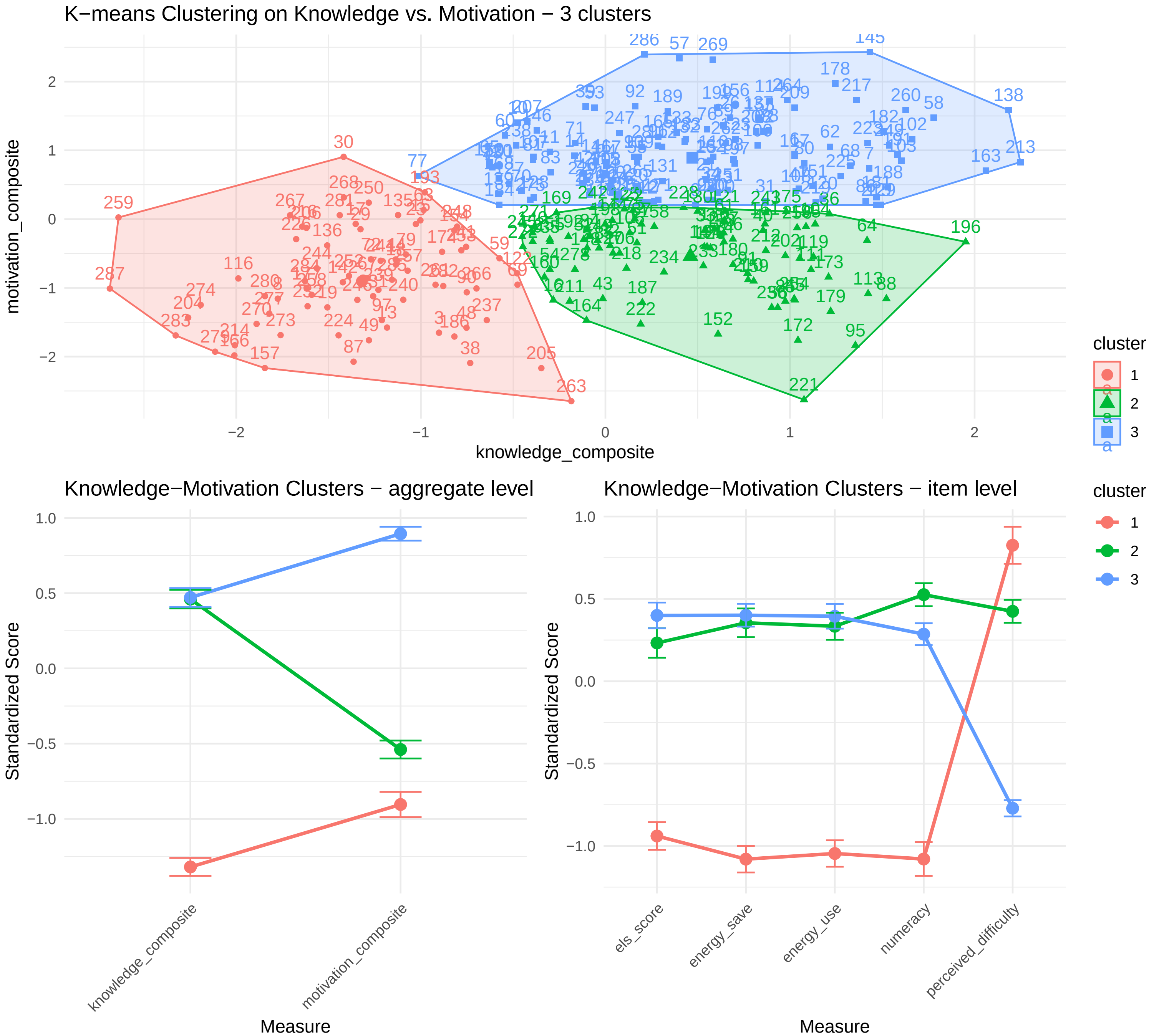
```
# Create composite scores for knowledge and motivation
combined_scores <- combined_scores %>%
  mutate(
    knowledge_composite = rowMeans(
      cbind(numeracy, energy_use, energy_save, els_score),
      na.rm = TRUE
    ),
    motivation_composite = rowMeans(
      cbind(env_attitude, -1 * perceived_difficulty),
      na.rm = TRUE
    )
  )

cluster_data <- combined_scores %>%
  select(knowledge_composite, motivation_composite) %>%
  na.omit() %>%
  scale()



# Determine the optimal number of clusters using the Elbow Method
fviz_nbclust(cluster_data, kmeans, method = "wss") +
  theme_minimal() +
  labs(title = "Elbow Method for Determining Optimal Number of Clusters")
```



3 clusters - Composite scores

```
# Decide the number of clusters (k). Let's try k = 3 for illustration:
set.seed(123)
km_fit <- kmeans(cluster_data, centers = 3, nstart = 25)
# Visualize clusters
p1 <- fviz_cluster(km_fit, data = cluster_data) +
  labs(title = "K-means Clustering on Knowledge vs. Motivation - 3 clusters") +
  theme_minimal()

# Create standardized scores for profile analysis
 profile_data <- combined_scores %>%
    mutate(cluster = factor(km_fit$cluster)) %>%
  select(id, knowledge_composite, motivation_composite, cluster) %>%
  gather(measure, value, -id, -cluster) %>%
  group_by(measure) %>%
  mutate(z_score = scale(value)[, 1]) %>%
  ungroup()

# Create profile plot
p2 <- ggplot(profile_data, aes(x = measure, y = z_score, color = cluster, group = cluster)) +
  stat_summary(fun = mean, geom = "line", size = 1) +
  stat_summary(fun = mean, geom = "point", size = 3) +
  stat_summary(fun.data = mean_se, geom = "errorbar", width = 0.2) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(
    title = "Knowledge-Motivation Clusters - aggregate level",
    x = "Measure",
    y = "Standardized Score"
  )

# Create standardized scores for profile analysis - using original item_columns "numeracy", "energy_use", "ene
 profile_data <- combined_scores %>%
    mutate(cluster = factor(km_fit$cluster)) %>%
  select(id, cluster, numeracy, energy_use, energy_save,
         els_score, perceived_difficulty) %>%
  gather(measure, value, -id, -cluster) %>%
  group_by(measure) %>%
  mutate(z_score = scale(value)[, 1]) %>%
```

```r
  ungroup()

# Create profile plot
p3 <- ggplot(profile_data, aes(x = measure, y = z_score, color = cluster, group = cluster)) +
  stat_summary(fun = mean, geom = "line", size = 1) +
  stat_summary(fun = mean, geom = "point", size = 3) +
  stat_summary(fun.data = mean_se, geom = "errorbar", width = 0.2) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(
    title = "Knowledge-Motivation Clusters - item level",
    x = "Measure",
    y = "Standardized Score"
  )

p1 / (p2 + p3) + plot_layout(guides = 'collect')
```




```
# Add cluster membership back to your main dataframe
combined_scores$km_cluster <- factor(km_fit$cluster)


# Compare mean knowledge & motivation by cluster
combined_scores %>%
  group_by(km_cluster) %>%
  summarise(
    mean_knowledge = mean(knowledge_composite, na.rm = TRUE),
    mean_motivation = mean(motivation_composite, na.rm = TRUE),
    n = n()
  ) |> kable()
```

km_cluster	mean_knowledge	mean_motivation	n
1	-1.04	1.1	75
2	0.36	1.4	85
3	0.37	2.5	127

```
# Summarize cluster profiles
cluster_profiles <- combined_scores %>%
  group_by(cluster) %>%
  summarise(
    n = n(),
    mean_knowledge = mean(knowledge_composite, na.rm=TRUE),
    sd_knowledge = sd(knowledge_composite, na.rm=TRUE),
    mean_motivation = mean(motivation_composite, na.rm=TRUE),
    sd_motivation = sd(motivation_composite, na.rm=TRUE)
  )
cluster_profiles |> kable()
```

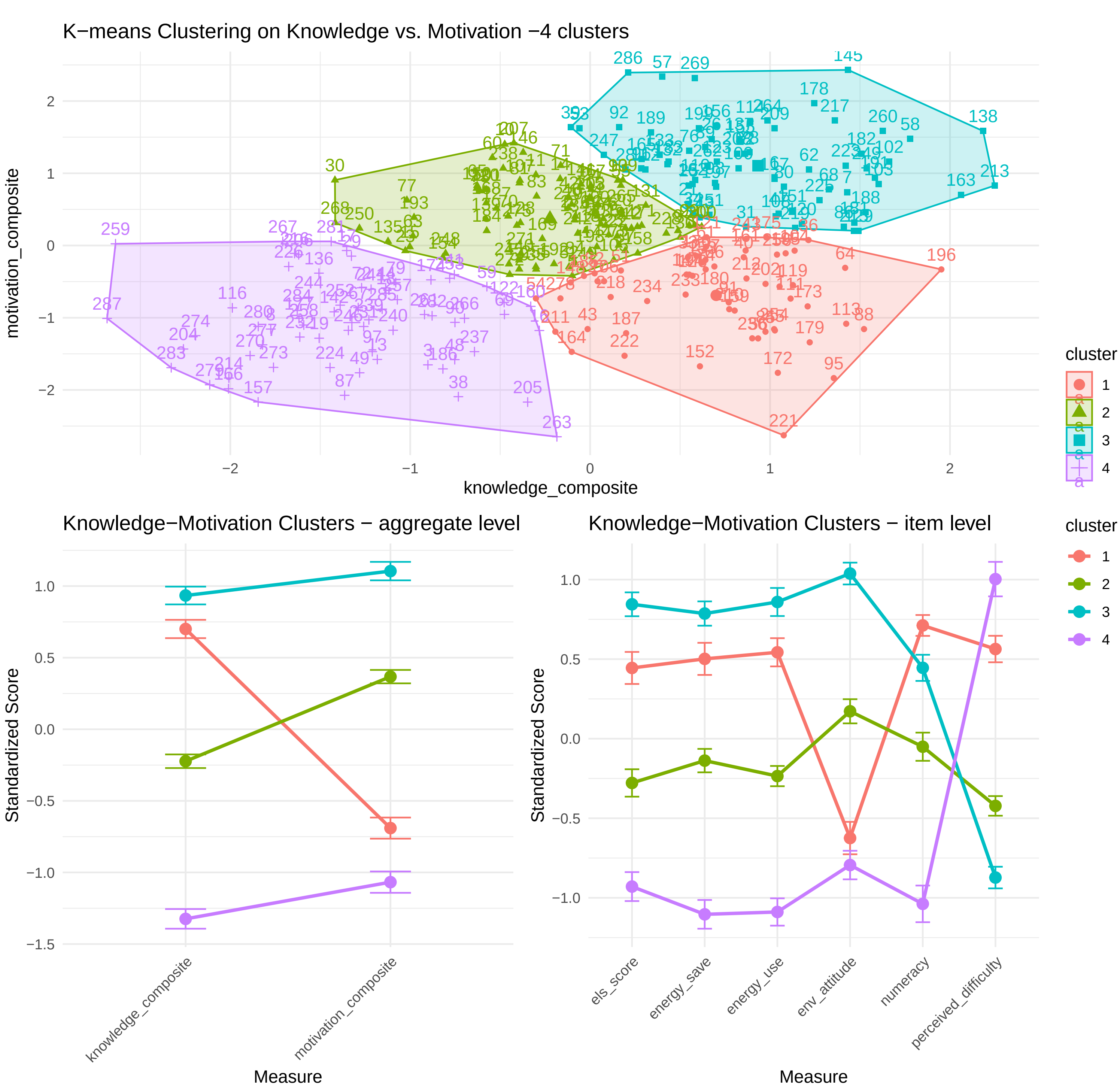
cluster	n	mean_knowledge	sd_knowledge	mean_motivation	sd_motivation
1	89	0.15	0.46	1.7	0.61
2	73	-1.06	0.39	1.1	0.56
3	125	0.51	0.50	2.3	0.59

4 clusters - Composite scores

```
# Decide the number of clusters (k). Let's try k =4 for illustration:
set.seed(123)
km_fit <- kmeans(cluster_data, centers = 4, nstart = 25)
# Visualize clusters
p1 <- fviz_cluster(km_fit, data = cluster_data) +
  labs(title = "K-means Clustering on Knowledge vs. Motivation -4 clusters") +
  theme_minimal()

# Create composite knowledge score
combined_scores$composite_knowledge <- rowMeans(combined_scores[, c("numeracy", "energy_use", "energy_save", "
combined_scores$cluster <- as.factor(km_fit$cluster)

# Create standardized scores for profile analysis
profile_data <- combined_scores %>%
```

```r
    mutate(cluster = factor(km_fit$cluster)) %>%
  select(id, knowledge_composite, motivation_composite, cluster) %>%
  gather(measure, value, -id, -cluster) %>%
  group_by(measure) %>%
  mutate(z_score = scale(value)[, 1]) %>%
  ungroup()

p2 <- ggplot(profile_data, aes(x = measure, y = z_score, color = cluster, group = cluster)) +
  stat_summary(fun = mean, geom = "line", size = 1) +
  stat_summary(fun = mean, geom = "point", size = 3) +
  stat_summary(fun.data = mean_se, geom = "errorbar", width = 0.2) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(
    title = "Knowledge-Motivation Clusters - aggregate level",
    x = "Measure",
    y = "Standardized Score"
  )

# Create standardized scores for profile analysis
profile_data <- combined_scores %>%
  select(id, cluster, numeracy, energy_use, energy_save,
         els_score, env_attitude, perceived_difficulty) %>%
  gather(measure, value, -id, -cluster) %>%
  group_by(measure) %>%
  mutate(z_score = scale(value)[, 1]) %>%
  ungroup()

p3 <- ggplot(profile_data, aes(x = measure, y = z_score, color = cluster, group = cluster)) +
  stat_summary(fun = mean, geom = "line", size = 1) +
  stat_summary(fun = mean, geom = "point", size = 3) +
  stat_summary(fun.data = mean_se, geom = "errorbar", width = 0.2) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(
    title = "Knowledge-Motivation Clusters - item level",
    x = "Measure", y = "Standardized Score"
  )
```

```r
p1 / (p2 + p3) + plot_layout(guides = 'collect')
```



This code creates composite scores for knowledge and motivation and then uses cluster analysis to identify distinct profiles based on these composite scores.

```r
# Add cluster membership back to your main dataframe
combined_scores$km_cluster <- factor(km_fit$cluster)


# Compare mean knowledge & motivation by cluster
combined_scores %>%
```

```r
  group_by(km_cluster) %>%
  summarise(
    mean_knowledge = mean(knowledge_composite, na.rm = TRUE),
    mean_motivation = mean(motivation_composite, na.rm = TRUE),
    n = n()
  ) |> kable()
```

km_cluster	mean_knowledge	mean_motivation	n
1	0.55	1.28	59
2	-0.18	2.07	89
3	0.73	2.63	72
4	-1.04	0.99	67

```r
# Summarize cluster profiles
cluster_profiles <- combined_scores %>%
  group_by(cluster) %>%
  summarise(
    n = n(),
    mean_knowledge = mean(knowledge_composite, na.rm=TRUE),
    sd_knowledge = sd(knowledge_composite, na.rm=TRUE),
    mean_motivation = mean(motivation_composite, na.rm=TRUE),
    sd_motivation = sd(motivation_composite, na.rm=TRUE)
  )
cluster_profiles |> kable()
```

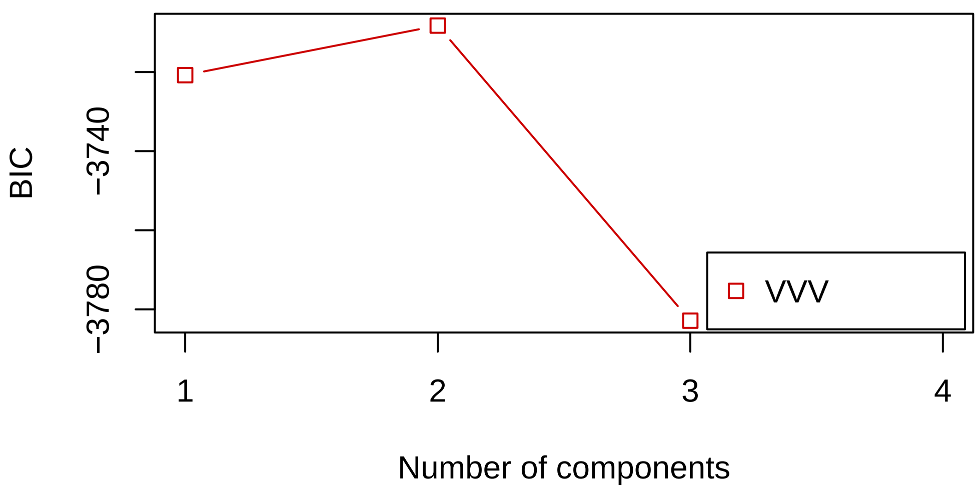
cluster	n	mean_knowledge	sd_knowledge	mean_motivation	sd_motivation
1	59	0.55	0.39	1.28	0.43
2	89	-0.18	0.35	2.07	0.33
3	72	0.73	0.41	2.63	0.41
4	67	-1.04	0.44	0.99	0.46

```r
# Select all motivation and knowledge scores for clustering
cluster_data <- combined_scores %>%
    select(perceived_difficulty, env_attitude, els_score, numeracy, energy_use)

# Perform model-based clustering using Mclust
# We'll let Mclust determine the optimal number of clusters (G = 1:4 as an example range, you can adjust)
mclust_result <- Mclust(cluster_data, G = 1:4, modelNames = "VVV") # VVV for variable variance, variable covar
```

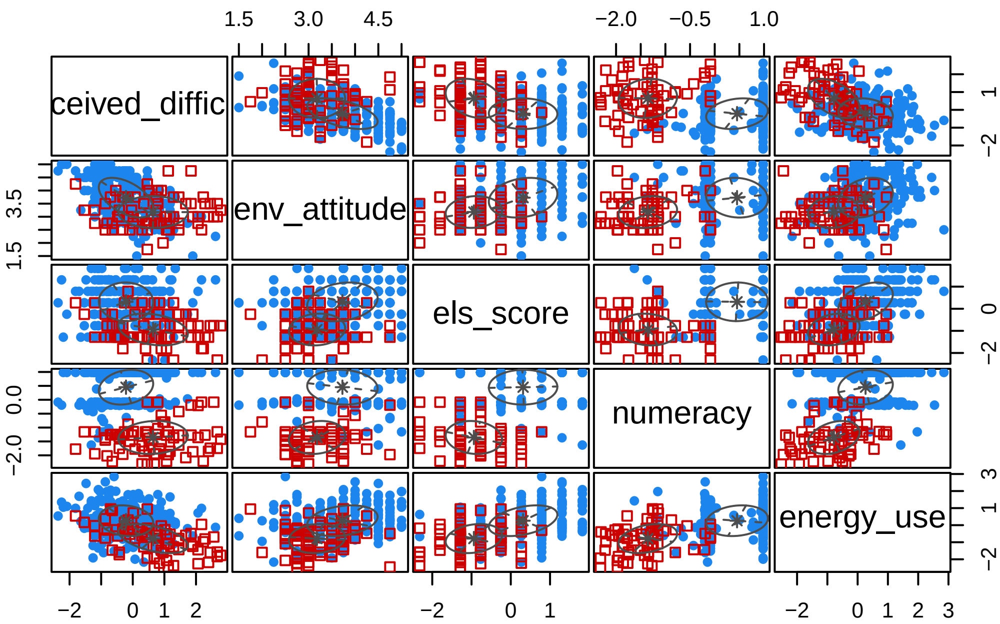
```r
# Summary of clustering
summary(mclust_result)
```

```
----------------------------------------------------
Gaussian finite mixture model fitted by EM algorithm
----------------------------------------------------

Mclust VVV (ellipsoidal, varying volume, shape, and orientation) model with 2
components:

 log-likelihood   n df   BIC   ICL
          -1738 287 41 -3708 -3725

Clustering table:
  1   2
216  71
```

```r
plot(mclust_result, what = "BIC") # BIC plot to help choose number of clusters
```




```r
plot(mclust_result, what = "classification") # Classification plot
```



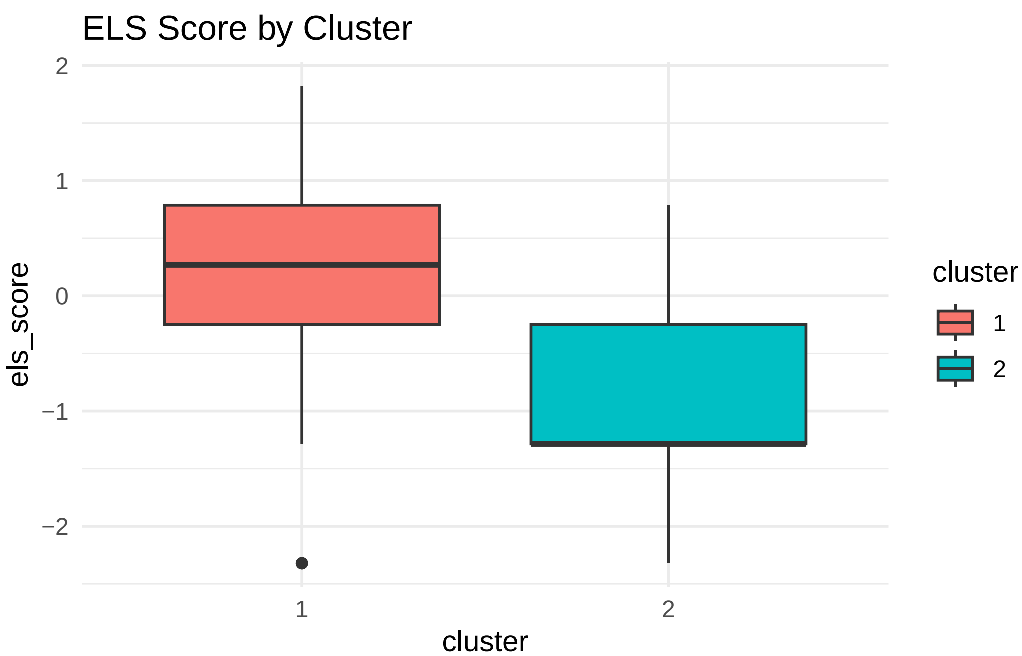

```r
# Get cluster assignments
cluster_assignments <- mclust_result$classification
combined_scores$cluster <- factor(cluster_assignments) # Add cluster assignments to your combined_scores data

# Analyze clusters - e.g., mean scores per cluster
cluster_means <- combined_scores %>%
    group_by(cluster) %>%
    summarise(
        mean_pd = mean(perceived_difficulty, na.rm = TRUE),
        mean_ea = mean(env_attitude, na.rm = TRUE),
        mean_els = mean(els_score, na.rm = TRUE),
        mean_num = mean(numeracy, na.rm = TRUE),
        mean_eu = mean(energy_use, na.rm = TRUE),
        n = n()
    )
print(cluster_means) |> kable()
```

```
# A tibble: 2 x 7
  cluster mean_pd mean_ea mean_els mean_num mean_eu     n
  <fct>     <dbl>   <dbl>    <dbl>    <dbl>   <dbl> <int>
1 1        -0.218    3.73    0.310    0.456   0.261   216
2 2         0.663    3.16   -0.942   -1.39   -0.793    71
```

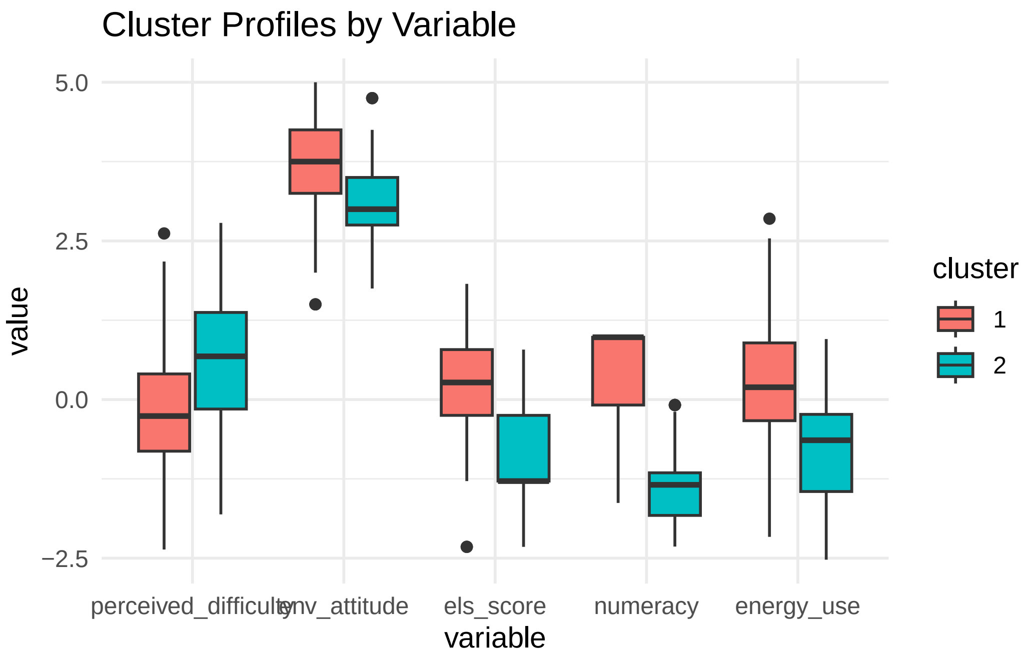
cluster	mean_pd	mean_ea	mean_els	mean_num	mean_eu	n
1	-0.22	3.7	0.31	0.46	0.26	216

cluster	mean_pd	mean_ea	mean_els	mean_num	mean_eu	n
2	0.66	3.2	-0.94	-1.39	-0.79	71

```
# Visualize clusters (e.g., using boxplots or profiles)
# Example boxplots for ELS score across clusters
ggplot(combined_scores, aes(x = cluster, y = els_score, fill = cluster)) +
    geom_boxplot() +
    labs(title = "ELS Score by Cluster") +
    theme_minimal()
```




```
# pivot long with perceived_difficulty, env_attitude, els_score, numeracy, energy_use - and create a faceted p

combined_scores_long <- combined_scores %>%
  pivot_longer(cols = c(perceived_difficulty, env_attitude, els_score, numeracy, energy_use),
               names_to = "variable", values_to = "value") |>
    mutate(cluster = factor(cluster)) |>
    mutate(variable = factor(variable, levels = c("perceived_difficulty", "env_attitude", "els_score", "numera
    mutate(cluster = factor(cluster, levels = c("1", "2", "3")))

ggplot(combined_scores_long, aes(x = variable, y = value, fill = cluster)) +
    geom_boxplot() +
   # facet_wrap(~variable, scales = "free_y") +
    labs(title = "Cluster Profiles by Variable") +
    theme_minimal()
```



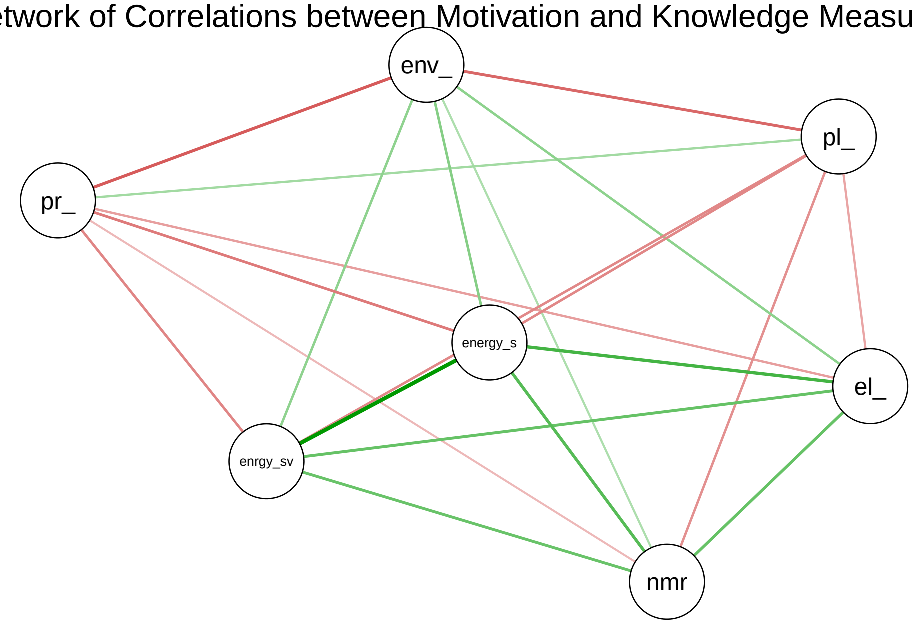

```r
# ANOVA to test for significant differences in means across clusters for each variable
variables_to_test <- c("perceived_difficulty", "env_attitude", "els_score", "numeracy", "energy_use")

anova_results <- list()
for (var in variables_to_test) {
  formula <- formula(paste(var, "~ cluster"))
  anova_model <- aov(formula, data = combined_scores)
  anova_results[[var]] <- summary(anova_model)
  cat("ANOVA for", var, ":\n")
  print(summary(anova_model))
  cat("\n")
}
```

```
ANOVA for perceived_difficulty :
             Df Sum Sq Mean Sq F value         Pr(>F)
cluster       1   41.4    41.4    48.3 0.000000000025 ***
Residuals   285  244.6     0.9
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

ANOVA for env_attitude :
             Df Sum Sq Mean Sq F value      Pr(>F)
cluster       1   17.3   17.33    33.2 0.000000022 ***
Residuals   285  148.9    0.52
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
ANOVA for els_score :
             Df Sum Sq Mean Sq F value              Pr(>F)
cluster       1   83.8    83.8     118 <0.0000000000000002 ***
Residuals   285  202.2     0.7
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

ANOVA for numeracy :
             Df Sum Sq Mean Sq F value              Pr(>F)
cluster       1    182   181.6     496 <0.0000000000000002 ***
Residuals   285    104     0.4
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

ANOVA for energy_use :
             Df Sum Sq Mean Sq F value              Pr(>F)
cluster       1   59.3    59.3    74.5 0.00000000000000044 ***
Residuals   285  226.7     0.8
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
    # Use the correlation matrix calculated earlier (cor_matrix)
if(!exists("cor_matrix")){ #recalculate if cor_matrix doesn't exist from previous code
    cor_matrix <- cor(combined_scores %>%
                        select(numeracy, energy_use, energy_save, els_score,
                                env_attitude, perceived_difficulty, pol_conservatism),
                        use = "pairwise.complete.obs")
}

qgraph(cor_matrix,
        graph = "cor", # Correlation graph
        layout = "spring", # Layout algorithm
        vsize = 8, # Vertex size
        esize = 3, # Edge size
        title = "Network of Correlations between Motivation and Knowledge Measures")
```

Network of Correlations between Motivation and Knowledge Measures



Clustering on question-level data

```
# combine question level data (aes_combined, att2_combined, els, rs,) by id

dq <- aes_combined |> left_join(att2_combined, by = "id") |> left_join(els, by = "id") |> left_join(rs, by = "




# Custom function for correlation matrix plots of question-level data
plot_cor_matrix_items <- function(data, title = NULL) {

cor_matrix <- cor(data, use = "pairwise.complete.obs")

# Identify item types based on column names
item_names <- colnames(cor_matrix)
is_attari_diff <- grepl("^ATT0[1-9]$|^ATT1[0-5]$", item_names) # ATT01-ATT15
is_attari_num <- grepl("^ATT1[6-8]$", item_names)  # ATT16-ATT18
is_attari_energy_use <- grepl("^ATT(19|2[0-7])$", item_names) # ATT19-ATT27
is_attari_energy_save <- grepl("^ATT(2[8-9]|3[0-3])$", item_names)  # ATT28-ATT33
is_els <- grepl("^ELS0[1-8]$", item_names) # ELS01-ELS08
is_rs <- grepl("^RS0[1-6]$", item_names) # RS01-RS06

item_groups <- ifelse(is_attari_diff, "Attari Difficulty",
                    ifelse(is_attari_num, "Attari Numeracy",
                            ifelse(is_attari_energy_use, "Attari Usage",
                                    ifelse(is_attari_energy_save, "Attari Savings",
                                        ifelse(is_els, "Energy Literacy",
                                                ifelse(is_rs, "Recycling Study", NA))))))

  qgraph(cor_matrix,
         layout = "spring",
         groups = list("Attari Difficulty" = which(item_groups == "Attari Difficulty"),
                       "Attari Numeracy" = which(item_groups == "Attari Numeracy"),
                       "Attari Usage" = which(item_groups == "Attari Usage"),
                       "Attari Savings" = which(item_groups == "Attari Savings"),
                       "Energy Literacy" = which(item_groups == "Energy Literacy"),
                       "Recycling Study" = which(item_groups == "Recycling Study")),
         color = c(rep("skyblue", sum(is_attari_diff)),
                  rep("lightcoral", sum(is_attari_num)),
```

```
                    rep("lightgreen", sum(is_attari_energy_use)),
                    rep("lightyellow", sum(is_attari_energy_save)),
                    rep("lightpink", sum(is_els)),
                     rep("lightcyan", sum(is_rs))),
          title = title)

}

plot_cor_matrix_items(dq |> select(-id))
```