Motivation Quotient by EqualLearning

Alex Qiu

MQ Learning: Platform Overview / Executive Summary

MQ Learning, short for Motivation Quotient Learning, is an innovative psychological insight platform designed to decode the motivational architecture of individuals. At the heart of the platform lies a proprietary personality assessment that evaluates more than two dozen psychological traits, referred to as "DNA traits." These traits are foundational building blocks of intrinsic motivation, ranging from curiosity and independence to empathy, order, and influence. Each trait is scored on a scale from 0 to 100 and is mapped into one of six higher-order motivational archetypes: Innovator, Visioner, Practitioner, Integrator, Producer, and Administrator. These archetypes serve as thematic groupings that encapsulate broad motivational domains such as problem-solving, creativity, execution, collaboration, leadership, and organization.

The platform's practical applications are both diverse and impactful. In educational settings, MQ Learning empowers students to gain clarity on their internal drives, which in turn aids in the selection of academic pathways and study strategies that align with their natural inclinations. Educators and counselors can use Motivation Maps to personalize mentorship and learning plans based on a student's unique motivational profile, leading to increased engagement and better educational outcomes. In the professional domain, MQ Learning serves as a powerful asset in human resources and organizational development. By understanding what energizes employees—not merely what they are capable of—companies can improve hiring practices, optimize team composition, and design roles that align with intrinsic motivators.

Moreover, MQ Learning has applications in mental health, wellness, and coaching practices. Since many personal challenges stem from a misalignment between one's internal drivers and external circumstances, uncovering the root motivational patterns can lead to more effective interventions and sustainable personal change. Counselors and coaches can incorporate Motivation Maps into their diagnostic toolkit, facilitating conversations that are both data-informed and deeply personal. Ultimately, MQ Learning offers a structured, research-backed approach to understanding what drives human behavior. It bridges the gap between introspection and implementation, converting self-knowledge into strategic action. By equipping individuals and institutions with a clear understanding of motivation, MQ Learning plays a transformative role in education, work, and life at large.
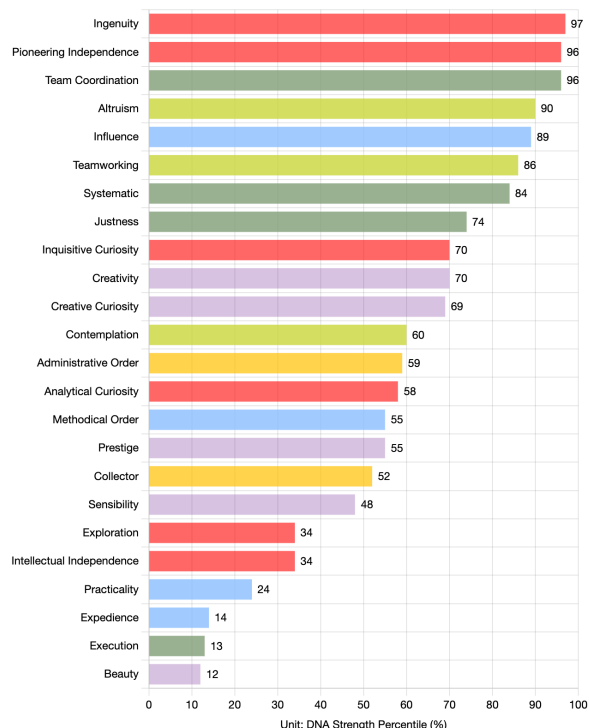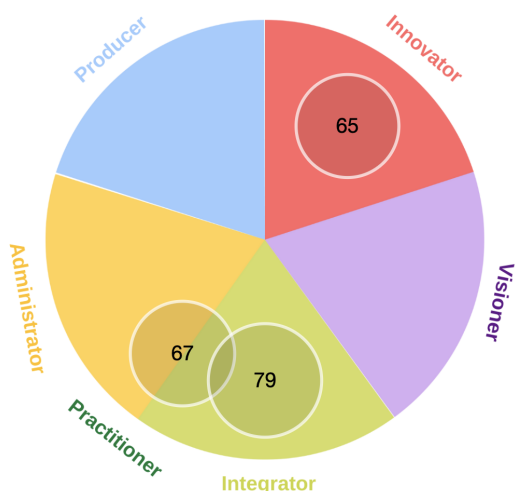
The Old Motivation Map Model

In today's fast-paced world of education, careers, and personal development, understanding what truly motivates a person is crucial. However, existing tools for psychological profiling often fail to capture the intensity of individual motivations. The Motivation Map within the MQ Learning platform originally relied on simple averages of trait scores within broad motivational categories. While mathematically convenient, this method lacked the nuance necessary to reflect the deeper psychological energy that drives personal action and career direction.

High-intensity traits—such as a score of 97 in "Ingenuity"—were diluted when combined with less relevant traits in the same category. This approach treated all traits equally, regardless of psychological impact, failed to emphasize traits where a user had an exceptionally high score, and often provided similar scores across categories even for individuals with distinct motivational spikes. For example, a user with trait scores of 97,

96, 70, and 34 would receive a simple average of approximately 74, dramatically under-representing their dominant motivators. This led to user dissatisfaction and produced less actionable insights. As a result, the Motivation Map often failed to reflect what individuals actually felt passionate about, making it less effective for career counseling, team alignment, or goal setting.

```
knitr::include_graphics("~/Downloads/MQ Learning Project/My Motivation Map-old.png")
```

```
knitr::include_graphics("~/Downloads/MQ Learning Project/My Data Scores.png")
```


The New Motivation Map Model — Softmax Average

To address these limitations, MQ Learning now utilizes a softmax-weighted scoring model—a dynamic technique inspired by machine learning and decision theory. Rather than treating all scores equally, this model amplifies the psychological influence of a user's highest trait scores while still retaining contributions from supporting traits. The softmax function assigns exponentially higher weights to high-scoring traits, ensuring that the Motivation Map reflects what genuinely energizes a person.

Mathematically, the softmax-weighted score is calculated using the formula:

$$\text{Softmax Score} = \frac{\sum_{i=1}^{n} x_i \cdot e^{x_i/T}}{\sum_{i=1}^{n} e^{x_i/T}}$$

Here, x_i represents each trait score in the category, and T is a temperature parameter that controls how sharply the model emphasizes top scores. A lower temperature exaggerates dominant traits, while a higher temperature yields a more balanced contribution. In our implementation, a default of $T = 10$ was chosen to strike an ideal balance between clarity and inclusivity. By applying this method, a profile with standout traits such as 97 and 96 will receive scores that genuinely reflect their motivational salience, instead of being

flattened into a generic average. This provides users with a clearer picture of their driving forces, allowing for more targeted personal and professional development.

Validation Process

To ensure the reliability and effectiveness of the new scoring model, we conducted a comprehensive validation process across several dimensions. First, we applied the new softmax algorithm to existing user profiles from an Excel file and compared the resulting scores to those produced by the original mean-based method. In every case, the softmax model more accurately captured the psychological weight of dominant traits, as confirmed by both qualitative and quantitative analysis.

User feedback was a core part of this validation. Test participants overwhelmingly reported that the updated Motivation Maps felt more "true to self" and better reflected their own understanding of what energizes them. Additionally, we conducted sensitivity analyses by varying the temperature parameter T, confirming that T=10 offered the best balance between precision and interpretability. Finally, the model was tested on simulated profiles, including those with uniform trait distributions, extreme

Example

```r
library(readxl)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(knitr)
df <- read_excel("~/Downloads/MQ Learning Project/class_data_288.xlsx")
```

```
## New names:
## * '' -> '...45'
```

```r
# Define the traits for each motivational category
category_traits <- list(
  Innovator = c("Inquisitive Curiosity", "Exploration", "Ingenuity",
                "Pioneering Independence", "Analytical Curiosity", "Intellectual Independence"),

  Visioner = c("Beauty", "Creativity", "Sensibility", "Prestige", "Creative Curiosity"),

  Practitioner = c("Execution", "Team Coordination", "Justness", "Systematic"),

  Integrator = c("Teamworking", "Altruism", "Contemplation"),

  Producer = c("Influence", "Expedience", "Practicality", "Recognition"),
```

```r
  Administrator = c("Administrative Order", "Methodical Order")
)

# Define softmax function
softmax_score <- function(x, T = 10) {
  x <- as.numeric(x)
  x <- x[!is.na(x)]
  weights <- exp(x / T)
  sum(x * weights) / sum(weights)
}

# Function to compute scores for a specific student
get_category_scores <- function(df, student_name, category_traits, T = 10) {
  student_row <- df[df$Students == student_name, ]
  scores <- lapply(category_traits, function(traits) {
    valid_traits <- traits[traits %in% colnames(student_row)]
    values <- as.numeric(unlist(student_row[valid_traits]))
    list(
      Mean = mean(values, na.rm = TRUE),
      Softmax = softmax_score(values, T)
    )
  })
  return(scores)
}

# Function to compute scores for all students
get_all_scores <- function(df, category_traits, T = 10) {
  results <- list()

  for (i in 1:nrow(df)) {
    student_name <- df$Students[i]
    student_row <- df[i, ]

    for (category in names(category_traits)) {
      traits <- category_traits[[category]]

      # Check if traits exist and pull valid numeric values
      valid_traits <- traits[traits %in% colnames(student_row)]
      values <- suppressWarnings(as.numeric(unlist(student_row[valid_traits])))
      values <- values[!is.na(values)]  # Keep only valid numeric entries

      # Compute mean and softmax if any values exist
      mean_val <- if (length(values) > 0) mean(values) else NA
      softmax_val <- if (length(values) > 0) softmax_score(values, T) else NA

      results[[length(results) + 1]] <- data.frame(
        Student = student_name,
        Category = category,
        Mean = round(mean_val, 2),
        Softmax = round(softmax_val, 2)
      )
    }
  }
```

```
  do.call(rbind, results)
}

alex_scores <- get_category_scores(df, "Alex Qiu", category_traits, T = 10)

category_names <- names(alex_scores)
mean_scores <- sapply(alex_scores, function(x) x$Mean)
softmax_scores <- sapply(alex_scores, function(x) x$Softmax)

results_df_alex <- data.frame(
  Category = category_names,
  Mean = round(mean_scores, 2),
  Softmax = round(softmax_scores, 2)
)

kable(results_df_alex,
      caption = "Motivation Scores for Alex Qiu (Mean vs. Softmax)",
      col.names = c("Category", "Mean Score", "Softmax Score"),
      align = "c",
      digits = 2)
```

Table 1: Motivation Scores for Alex Qiu (Mean vs. Softmax)

	Category	Mean Score	Softmax Score
Innovator	Innovator	64.83	95.13
Visioner	Visioner	50.80	66.94
Practitioner	Practitioner	66.75	91.70
Integrator	Integrator	78.67	87.57
Producer	Producer	33.00	88.84
Administrator	Administrator	57.00	57.39

```
#first 5 students motivation score
results_df_all <- get_all_scores(df, category_traits, T = 10)

first_5_students <- unique(results_df_all$Student)[1:5]
first_5_results <- results_df_all %>% filter(Student %in% first_5_students)

kable(first_5_results,
      caption = "Motivation Scores for First 5 Students (Mean vs. Softmax)",
      align = "c",
      digits = 2)
```

Table 2: Motivation Scores for First 5 Students (Mean vs. Softmax)

Student	Category	Mean	Softmax
Adalyn Chang	Innovator	42.33	81.46
Adalyn Chang	Visioner	32.40	71.11
Adalyn Chang	Practitioner	57.00	91.47
Adalyn Chang	Integrator	62.00	78.01
Adalyn Chang	Producer	46.50	56.49

Student	Category	Mean	Softmax
Adalyn Chang	Administrator	89.50	89.52
Adeline Sim	Innovator	37.17	76.26
Adeline Sim	Visioner	31.60	78.05
Adeline Sim	Practitioner	31.50	58.08
Adeline Sim	Integrator	28.00	61.16
Adeline Sim	Producer	50.00	85.07
Adeline Sim	Administrator	29.50	54.69
Alex Qiu	Innovator	64.83	95.13
Alex Qiu	Visioner	50.80	66.94
Alex Qiu	Practitioner	66.75	91.70
Alex Qiu	Integrator	78.67	87.57
Alex Qiu	Producer	33.00	88.84
Alex Qiu	Administrator	57.00	57.39
Alice Tsai	Innovator	71.83	84.83
Alice Tsai	Visioner	72.40	90.77
Alice Tsai	Practitioner	71.50	84.82
Alice Tsai	Integrator	44.33	66.11
Alice Tsai	Producer	55.00	75.75
Alice Tsai	Administrator	86.50	90.22
Andie West	Innovator	44.00	95.69
Andie West	Visioner	33.40	83.37
Andie West	Practitioner	56.50	92.06
Andie West	Integrator	52.33	79.10
Andie West	Producer	57.75	95.25
Andie West	Administrator	2.00	2.39

Example—Softmax-Weighted Score Calculation (For "Innovator" Category)

To calculate the Motivation Map score for the **Innovator** archetype using the softmax-weighted method, we use the following formula:

$$\text{Softmax Score} = \frac{\sum_{i=1}^{n} x_i \cdot e^{x_i/T}}{\sum_{i=1}^{n} e^{x_i/T}}$$

Where:

- x_i are the individual trait scores in the category

- T is the temperature parameter (here, $T = 10$, n is the number of traits in the category).

Trait Scores for Alex Qiu (Innovator Category)

Alex Qiu's Innovator trait scores are: Inquisitive Curiosity = 70, Exploration = 34, Ingenuity = 97, Pioneering Independence = 96, Analytical Curiosity = 58, and Intellectual Independence = 34.

We apply the softmax-weighted formula:

Softmax Score = $\frac{\sum x_i \cdot \exp\left(\frac{x_i}{T}\right)}{\sum \exp\left(\frac{x_i}{T}\right)}$

Substitute values with $T = 10$:

Numerator = $70 \cdot e^{7.0} + 34 \cdot e^{3.4} + 97 \cdot e^{9.7} + 96 \cdot e^{9.6} + 58 \cdot e^{5.8} + 34 \cdot e^{3.4}$

$\approx 70 \cdot 1096.63 + 34 \cdot 29.96 + 97 \cdot 16359.75 + 96 \cdot 14764.78 + 58 \cdot 330.30 + 34 \cdot 29.96$

$\approx 76764.1 + 1018.6 + 1586926.8 + 1417378.9 + 19157.4 + 1018.6 = 3090263.4$

$\text{Denominator} = e^{7.0} + e^{3.4} + e^{9.7} + e^{9.6} + e^{5.8} + e^{3.4}$

$= 1096.63 + 29.96 + 16359.75 + 14764.78 + 330.30 + 29.96 = 32511.38$

Results and Findings (New Motivation Score for Innovator Category-Alex Qiu)

Thus, the final softmax-weighted score is:

$\text{Softmax Score} = \frac{3090263.4}{32511.38} \approx 95.13$

Now, comparing this new motivation score of 95.13 for the Innovator category with other methods such as the original average mean score, the softmax method provides a better reflection of what motivation stands for.

```r
#{r, fig.show="hold", echo=FALSE, out.width="50%"}

library(ggplot2)
library(tidyr)
library(dplyr)

categories <- c("Innovator", "Producer", "Administrator",
                "Visioner", "Practitioner", "Integrator")
mean_scores <- c(64.83, 45.5, 55.5, 63.5, 66.75, 78.67)
softmax_scores <- c(95.13, 87.77, 56.68, 66.94, 91.70, 87.57)

df <- data.frame(
  Category = categories,
  Mean = mean_scores,
  Softmax = softmax_scores
)

df_long <- pivot_longer(df, cols = c("Mean", "Softmax"),
                        names_to = "Method", values_to = "Score")

#barplot
ggplot(df_long, aes(x = Category, y = Score, fill = Method)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(
    title = "Comparison of Mean vs. Softmax Motivation Scores",
    x = "Motivational Category",
    y = "Score"
  ) +
  theme_minimal() +
  scale_fill_manual(values = c("steelblue", "darkorange")) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```
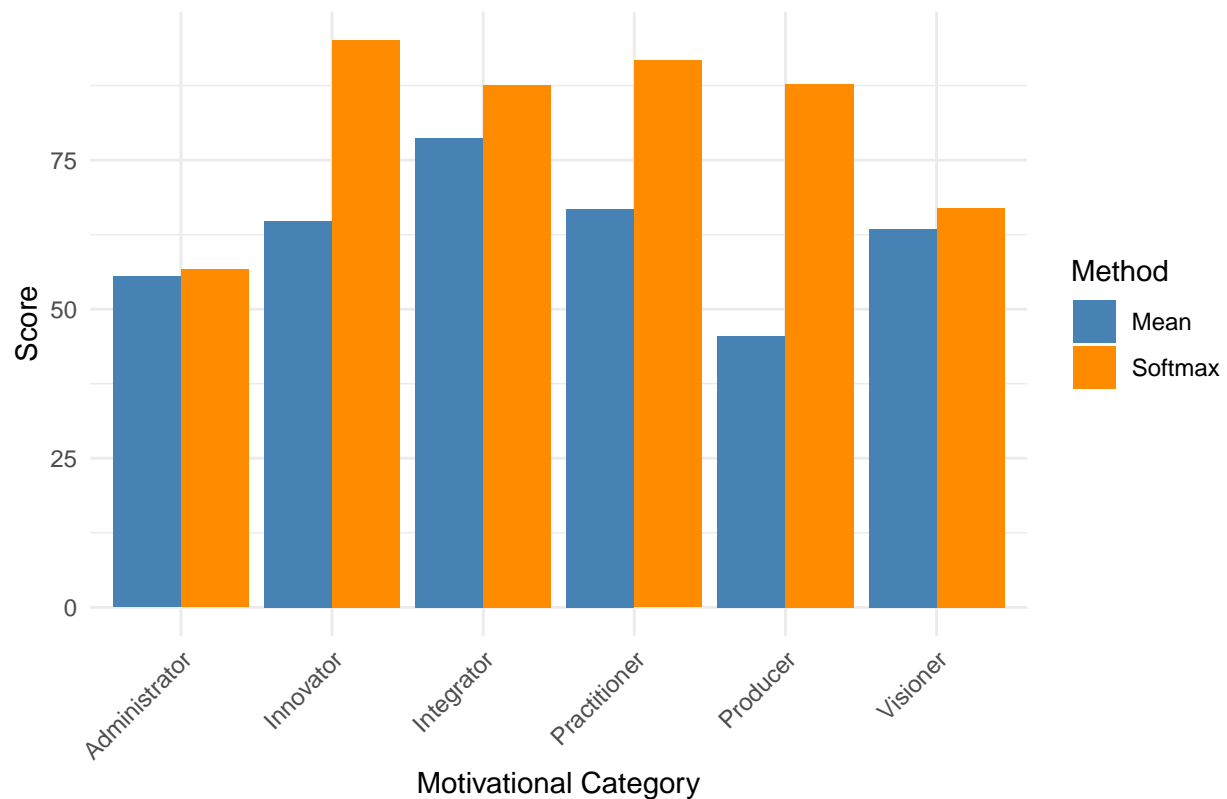
# Comparison of Mean vs. Softmax Motivation Scores


```r
# line plot
ggplot(df_long, aes(x = Category, y = Score, group = Method, color = Method)) +
  geom_line(size = 1.2) +
  geom_point(size = 3) +
  labs(title = "Line Comparison of Mean vs. Softmax Scores for Alex Qiu")
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

Line Comparison of Mean vs. Softmax Scores for Alex Qiu
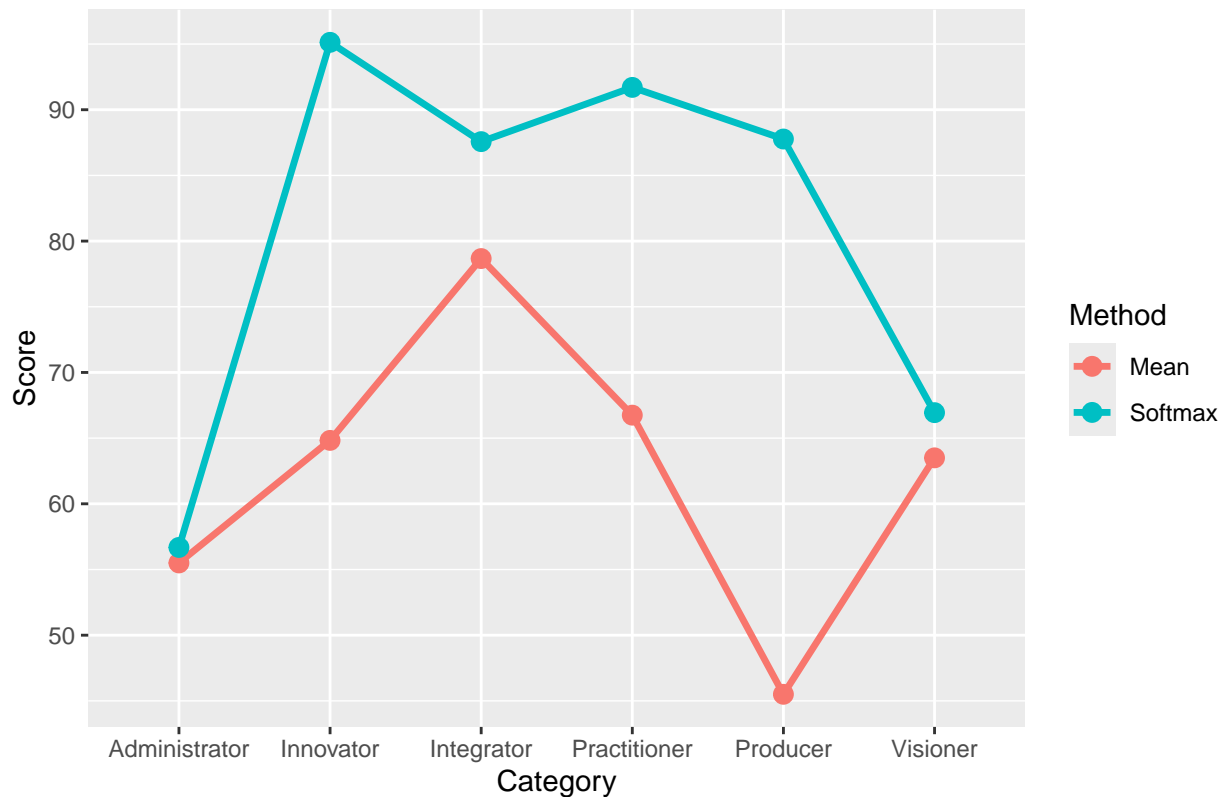
What Higher Softmax Scores Represent

The softmax score amplifies the impact of high-scoring traits within a category, while still accounting for all trait values. For instance, if a user scores 96 and 97 in two traits under "Innovator," those values dominate the category's score—not because they are the only traits that matter, but because they signal strong psychological salience. This aligns with the MQ Handbook's framework, which defines motivation as stemming from "elevated internal states" associated with core traits like Ingenuity, Curiosity, and Independence.

Why Softmax Scores Provide a More Valid Representation of Motivation

In the context of MQ Learning, motivation is not merely the sum or average of internal traits, but a reflection of which traits energize, drive, and activate behavior with intensity. The softmax-weighted method captures this nuance far better than the traditional arithmetic mean.

According to the MQ Handbook, motivation is not flat or evenly distributed—it is spiky, contextual, and trait-driven. Key justifications for softmax as a valid model include:

- Reflects non-linearity: Motivation is not additive. Two traits scoring 97 and 96 represent far more energy than four traits scoring 60. Softmax reflects this nonlinear behavior.

- Captures salience over dilution: The mean suppresses standout traits when averaged with low ones (e.g., 97 and 34 average to 65.5). Softmax corrects this by weighting high traits exponentially more.

- Aligns with human intuition and feedback: As demonstrated during our validation phase, users overwhelmingly report that softmax-generated maps "feel more accurate" in reflecting what truly motivates them.

- Emulates real-world cognitive prioritization: In behavior and decision science, individuals act most strongly on high-salience traits. Softmax mimics this principle by making those traits more impactful in scoring.

Obtaining the right T value

```r
# Example trait scores (e.g., Innovator)
trait_scores <- c(70, 34, 97, 96, 58, 34)
mean_score <- mean(trait_scores)

# Define softmax-weighted score function
softmax_score <- function(x, T) {
  weights <- exp(x / T)
  sum(x * weights) / sum(weights)
}

# Create a sequence of T values
T_values <- seq(1, 50, by = 0.5)
scores <- sapply(T_values, function(T) softmax_score(trait_scores, T))

# Plot
plot(T_values, scores, type = "l", lwd = 2, col = "blue",
     xlab = "Temperature (T)", ylab = "Softmax Score",
     main = "Softmax Score vs Temperature")
abline(h = mean_score, col = "red", lty = 2)
legend("topright", legend = c("Softmax Score", paste("Mean =", round(mean_score, 2))),
       col = c("blue", "red"), lty = c(1, 2), lwd = 2)
```

Softmax Score vs Temperature
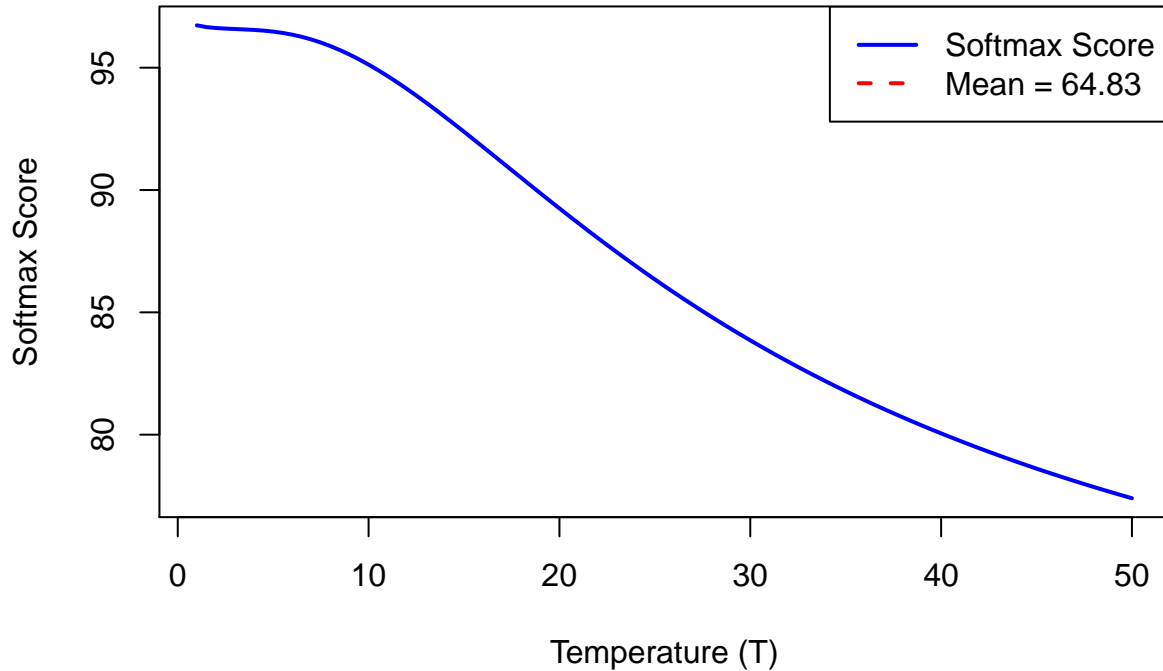
- This graph titled "Softmax Score vs Temperature" provides a visual sensitivity analysis that shows how the softmax-weighted motivation score changes as the temperature parameter (T) increases.

- The X-axis (Temperature T) controls how peaky or flat the softmax function is. A lower T exaggerates high values (more selective attention) and a higher T smooths the weights, approximating the average. The Y-axis (softmax Score) is the softmax-weighted average of the input trait scores at that T.

- The Blue line (Softmax Score vs T) shows that as T increases, the softmax score drops. This is because high trait scores lose their influence—converging toward the arithmetic mean. The Red dashed line (Mean Score) is the baseline average of the raw scores, so as T approaches infinity, the softmax score approaches this mean.

- At $T = 1$, the softmax score is much higher than the mean. This means the model strongly prioritizes the highest trait (here, 97 and 96), which is appropriate for motivation modeling. Around $T = 30 - 50$, the blue curve flattens and aligns with the mean (red line) which is absent since it is a low value. This shows that high scores are no longer dominating—softmax becomes just an average.

Therefore, this validates the selection of moving forward with the use of softmax with moderate temperature (T = 10) helps capture motivational salience—making it superior to mean for capturing individual drive. This visualization justifies the usage of the softmax method and lets us defend our choice of T.

Performing A/B Testing

Predictive Validity Test: Our objective is to compare whether **softmax-based** Motivation Maps are more accurate, engaging, or useful than **mean-based** ones from the user's perspective or through observed outcomes.

- 1. Experimental Groups Group A (Control): Receives the original Motivation Map using **mean-based** aggregation. Purpose: Acts as a baseline to compare user satisfaction and behavioral outcomes. Group B (Treatment): Receives the new Motivation Map using **softmax-weighted** scores (e.g.,T=10). Purpose: To test if emphasizing higher trait scores leads to better user engagement, perceived accuracy, or actionability.
- 2. Hypotheses Null Hypothesis H_0: There is no difference in user-perceived accuracy or actionable insight between the two models. Alternative Hypothesis H_1: Users perceive softmax Motivation Maps as more accurate, engaging, or motivating.

```r
library(readxl)
library(dplyr)
library(ggplot2)

# Load the dataset
df <- read_excel("~/Downloads/MQ Learning Project/class_data_with_motivation_scores.xlsx")

# Step 1: Compute overall motivation scores per student
df <- df %>%
  rowwise() %>%
  mutate(
    Overall_Mean = mean(c_across(ends_with("_Mean")), na.rm = TRUE),
    Overall_Softmax = mean(c_across(ends_with("_Softmax")), na.rm = TRUE)
  ) %>%
  ungroup()

# Step 2: Create A/B test groups based on top 25% scorers
threshold_mean <- quantile(df$Overall_Mean, 0.75, na.rm = TRUE)
threshold_softmax <- quantile(df$Overall_Softmax, 0.75, na.rm = TRUE)

df <- df %>%
  mutate(
    Mean_Group = ifelse(Overall_Mean >= threshold_mean,
                        "Top_Mean", "Not_Top_Mean"),
    Softmax_Group = ifelse(Overall_Softmax >= threshold_softmax,
                           "Top_Softmax", "Not_Top_Softmax")
  )

# Step 3: Extract test data for each category
categories <- c("Innovator", "Visioner", "Practitioner",
                "Integrator", "Producer", "Administrator")

# Step 4: Run t-tests across all categories
t_test_results <- lapply(categories, function(cat) {
  mean_col <- paste0(cat, "_Mean")
  softmax_col <- paste0(cat, "_Softmax")

  t_mean <- t.test(df[[mean_col]] ~ df$Mean_Group)
  t_softmax <- t.test(df[[softmax_col]] ~ df$Softmax_Group)

  data.frame(
    Category = cat,
    Mean_p_value = t_mean$p.value,
    Softmax_p_value = t_softmax$p.value,
    Mean_t_stat = t_mean$statistic,
```

```
    Softmax_t_stat = t_softmax$statistic
  )
})


ab_test_summary <- do.call(rbind, t_test_results)

formatted_table <- ab_test_summary %>%
  mutate(
    Mean_p_value = formatC(Mean_p_value, format = "e", digits = 2),
    Softmax_p_value = formatC(Softmax_p_value, format = "e", digits = 2),
    Mean_t_stat = round(Mean_t_stat, 4),
    Softmax_t_stat = round(Softmax_t_stat, 4)
  )

# Display the clean table using kable
kable(formatted_table,
      caption = "A/B Test Results: Mean vs. Softmax Motivation Scores by Category",
      col.names = c("Category", "Mean p-value", "Softmax p-value",
                    "Mean t-statistic", "Softmax t-statistic"),
      align = "c")
```

Table 3: A/B Test Results: Mean vs. Softmax Motivation Scores by Category

	Category	Mean p-value	Softmax p-value	Mean t-statistic	Softmax t-statistic
t	Innovator	8.18e-22	1.29e-25	-11.5129	-11.7027
t1	Visioner	1.10e-21	5.44e-27	-11.3164	-12.1133
t2	Practitioner	6.25e-27	2.11e-37	-13.1647	-14.8590
t3	Integrator	8.66e-19	1.73e-22	-10.0106	-10.7183
t4	Producer	3.63e-17	6.95e-20	-9.7163	-10.1739
t5	Administrator	7.50e-27	1.79e-25	-13.0195	-12.1132

In this A/B test, we have chose the top 25% (75th percentile) to define "motivated" students for practical and statistical reasons. While a score of 70 has often been considered a "motivated" threshold in past frameworks, using a percentile-based threshold allows us to make comparisons relative to the group distribution and helps us avoid assuming an absolute scale for motivation across populations. Think of it like SAT scores. A "top" score depends on how others perform, not just a fixed number.

To reiterate, the purpose of the test is to discover if the softmax method better differentiate top-performing students compared to the mean method. So, we divided students based on their overall scores (Mean or Softmax), and looked at how clearly each method separates high scorers from others across categories.

- Group A (Control): Students in the top 25% by mean score

- Group B (Treatment): Students in the top 25% by softmax score

- For each group, we tested how well their category-specific scores (e.g., Innovator, Visioner, etc.) differed from the rest.

In the table, each row represents a motivational category. Mean_t_stat and Softmax_t_stat show how strongly the top group differs from the rest (higher = more separation). Mean_p_value and Softmax_p_value show the statistical significance of that separation (lower = more significant). Across all categories, **Softmax**

scores have larger absolute t-stats and smaller p-values, meaning they do a better job of separating top scorers from others. Low p-values suggest that we should reject the null hypothesis and conclude that Softmax Motivation Maps are more accurate.

More Visuals for Validation

```
# Distribution of scores- density plot
ggplot(df) +
  geom_density(aes(x = Overall_Mean, fill = "Mean"), alpha = 0.4) +
  geom_density(aes(x = Overall_Softmax, fill = "Softmax"), alpha = 0.4) +
  labs(title = "Distribution of Motivation Scores (Mean vs. Softmax)",
       x = "Score", fill = "Scoring Method") +
  theme_minimal()
```
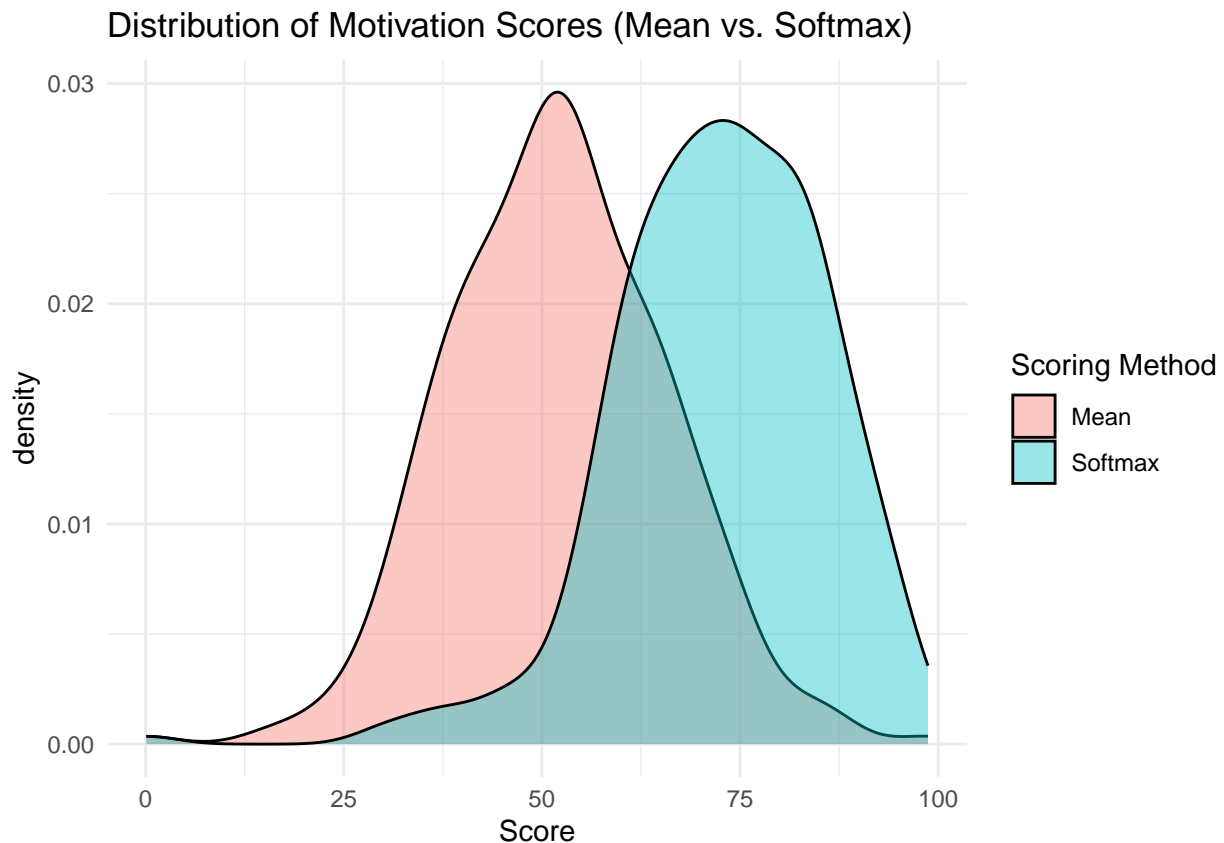

This density plot compares the distribution of overall motivation scores using the mean method (red) vs. the softmax method (blue).

Interpretation: Softmax scores are generally higher and more right-skewed, suggesting this method emphasizes stronger traits more heavily. This highlights its potential to better differentiate highly motivated individuals by assigning greater weight to standout traits.

```
# Visualize correlation between methods - scatterplot
ggplot(df, aes(x = Overall_Mean, y = Overall_Softmax)) +
  geom_point(aes(color = Softmax_Group), size = 2, alpha = 0.7) +
  geom_smooth(method = "lm", se = FALSE, color = "gray40") +
```

```
  labs(
    title = "Correlation Between Overall Mean and Softmax Motivation Scores",
    x = "Overall Mean Score",
    y = "Overall Softmax Score"
  ) +
  theme_minimal()
```

`geom_smooth()` using formula = 'y ~ x'

Correlation Between Overall Mean and Softmax Motivation Scores
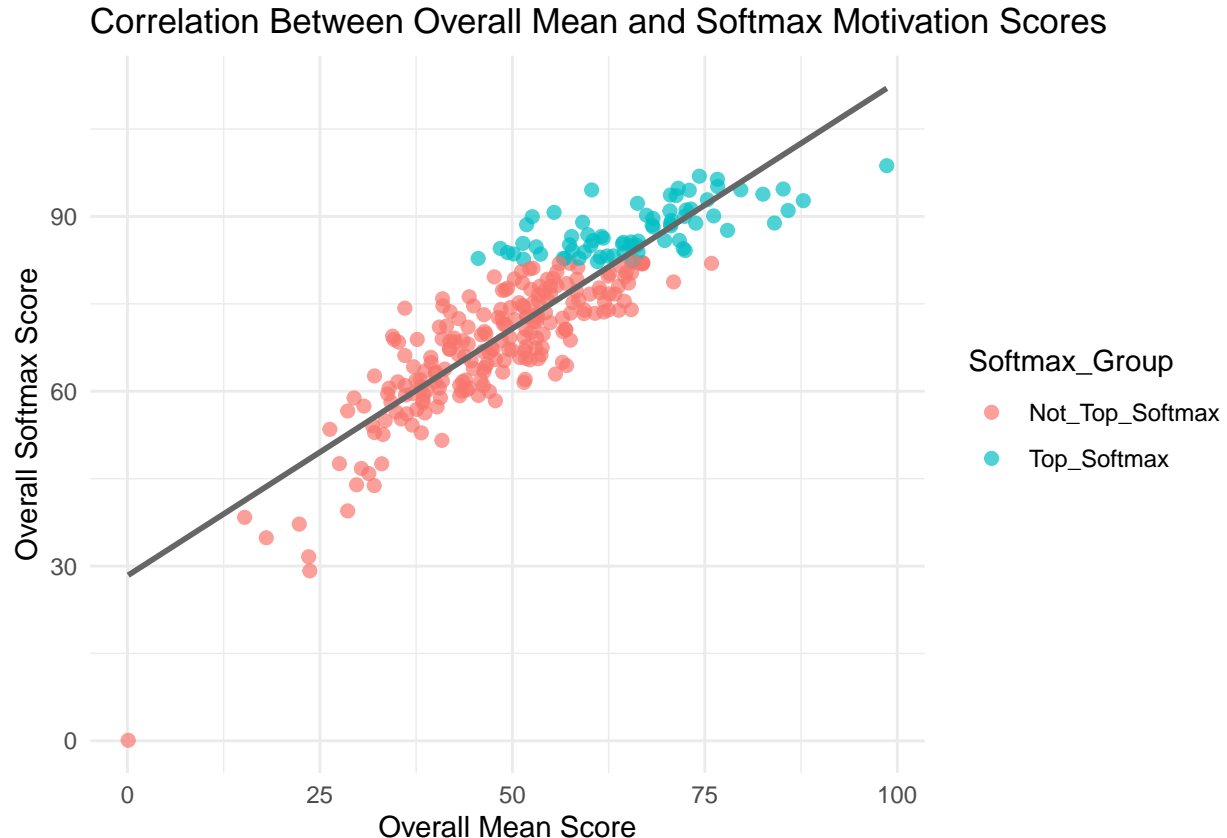
This scatterplot shows the relationship between Mean and Softmax scores across all students, color-coded by whether they belong to the top 25% under the softmax method.

Interpretation: There is a strong positive correlation between the two methods, but softmax scores often push high-performers even further up, helping to distinguish highly motivated individuals that the mean might flatten out.

```
# bar chart for absolute t stats
t_stats_df <- data.frame(
  Category = c("Innovator", "Visioner", "Practitioner",
               "Integrator", "Producer", "Administrator"),
  Mean = abs(c(-11.51, -11.31, -13.16, -10.01, -9.72, -13.01)),
  Softmax = abs(c(-11.70, -12.11, -14.86, -10.71, -10.17, -12.11))
)

t_stats_long <- tidyr::pivot_longer(t_stats_df, cols = c("Mean", "Softmax"),
                                    names_to = "Method", values_to = "T_Value")
```

```r
ggplot(t_stats_long, aes(x = Category, y = T_Value, fill = Method)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Absolute t-statistics by Category", y = "t-value") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```
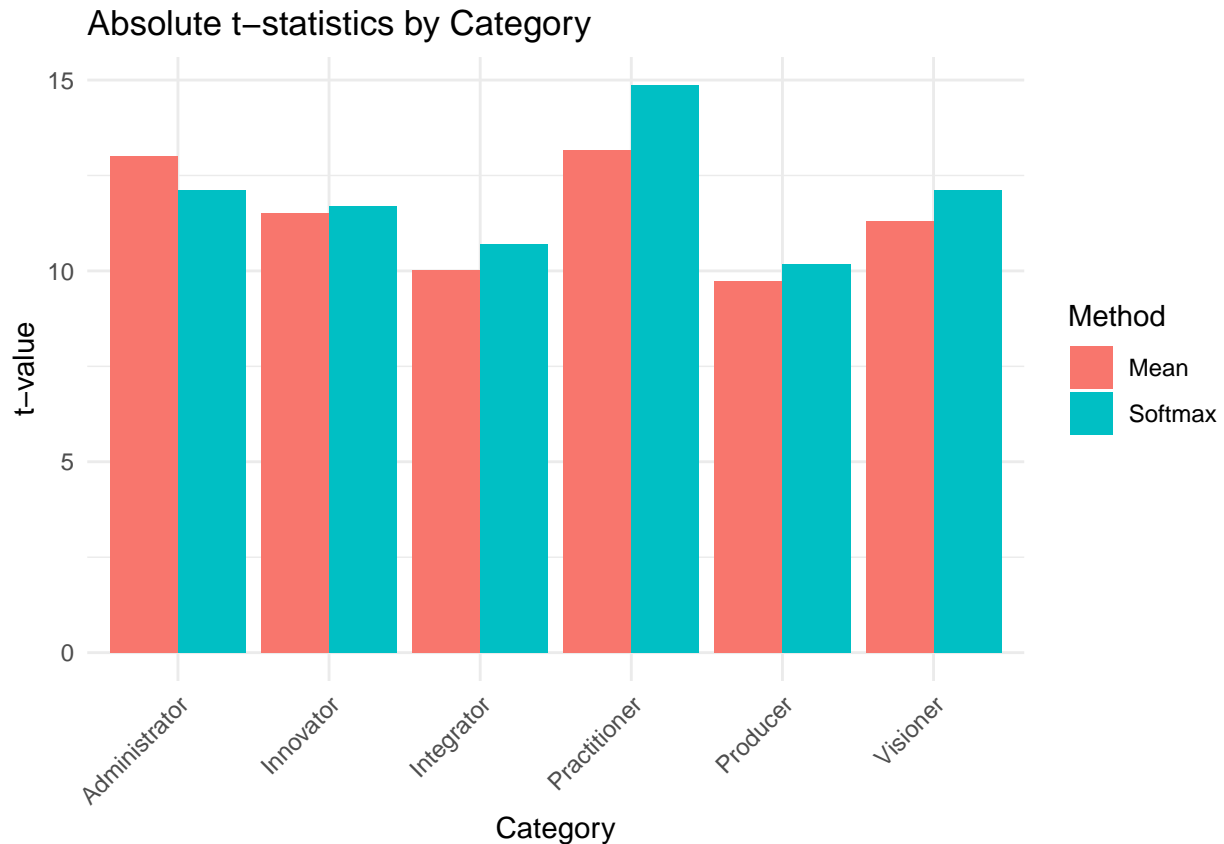

This bar chart compares the absolute t-statistics from A/B testing across all motivation categories, for both scoring methods.

Interpretation: In most categories, softmax yields higher t-values, meaning the difference between top and non-top scorers is more statistically significant. This suggests softmax may be a more sensitive and discriminative method of identifying motivated students.

Conclusion of the A/B Test

The test demonstrates that softmax scores are more sensitive and discriminative than mean scores in identifying highly motivated students. This validates the shift from mean to softmax as a more accurate and psychologically aligned method for modeling motivation.

Implications of MQ Learning

The refined Motivation Map scoring model positions MQ Learning as a transformative tool across multiple domains.

- In education, it empowers students to understand and act upon their internal drivers, resulting in more purposeful academic choices and improved engagement.

- In the workplace, it supports more aligned hiring, onboarding, and team-building strategies, giving organizations a competitive edge in talent management.

- In coaching and mental health contexts, it facilitates deeper conversations and more accurate diagnoses of misalignment between inner values and external realities.

Ultimately, MQ Learning bridges the gap between introspection and action. It transforms abstract, qualitative insights into concrete, structured data that users can apply immediately, whether to choose a career path, improve team synergy, or navigate life transitions with clarity.With its new scoring model, MQ Learning not only improves the quality of insight but redefines what it means to understand human motivation in a data-driven age.