

Rube This Cube: Self-Report Analysis

aiswary-a

2025-01-02

Loading In: Packages & CSV

```
library(pacman)
pacman::p_load(tidyverse)

overlord <- read_csv(
  "C:/Users/aisro/Desktop/UNI/BSc CogSci - 3rd Sem/Perception and Action/EXAM_Rubiks/PA_RubeCube_Experi
)

## New names:
## Rows: 9 Columns: 13
## -- Column specification
## ----- Delimiter: "," chr
## (6): Timestamp, RESEARCHER ONLY: Condition, Consent, Please supply your ... dbl
## (7): How old are you?, How confident are you about your ability to succe...
## i Use 'spec()' to retrieve the full column specification for this data. i
## Specify the column types or set 'show_col_types = FALSE' to quiet this message.
## * 'How difficult do you find this learning process to be for you?' -> 'How
##   difficult do you find this learning process to be for you?...10'
## * 'How difficult do you find this learning process to be for you?' -> 'How
##   difficult do you find this learning process to be for you?...12'

df <- overlord[, c(2, 4, 5, 6, 7, 8, 9, 10, 11, 12)]
colnames(df) <- c(
  "condition",
  "id",
  "gender",
  "age",
  "1_confidence",
  "1_difficulty",
  "2_confidence",
  "2_difficulty",
  "3_confidence",
  "3_difficulty",
  "performance"
)

df <- df %>%
  filter(id != "SR08")
```

Restructuring Data for Plotting

```
df_long <- df %>%
  pivot_longer(
    cols = starts_with(c("1_", "2_", "3_")),
    names_to = c("time", "metric"),
    names_pattern = "(\\d)_(.*)",
    values_to = "score"
  )

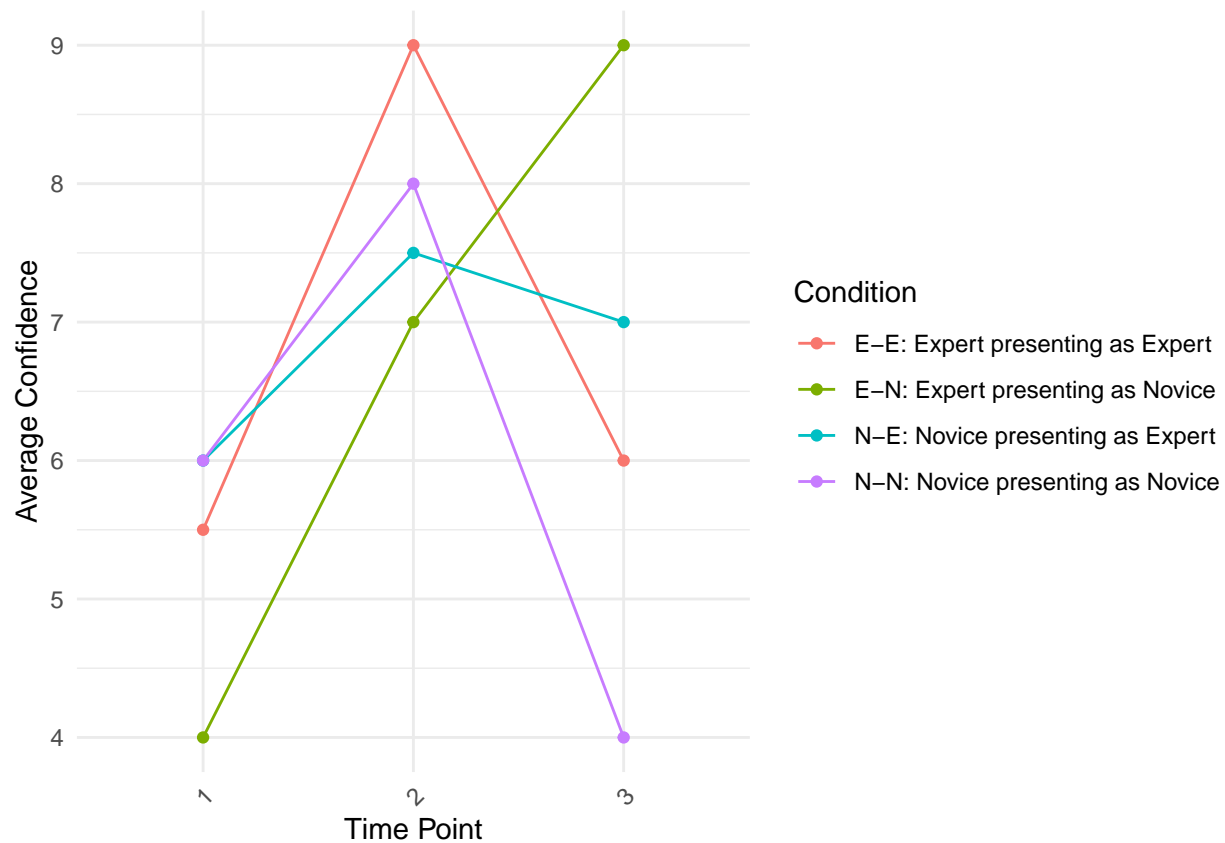
df_summary <- df_long %>%
  group_by(condition, time, metric) %>%
  summarise(
    mean = mean(score, na.rm = TRUE),
    se = sd(score, na.rm = TRUE) / sqrt(n())
  )
```

'summarise()' has grouped output by 'condition', 'time'. You can override using
the '.groups' argument.

Plotting Self-Assessments over Time

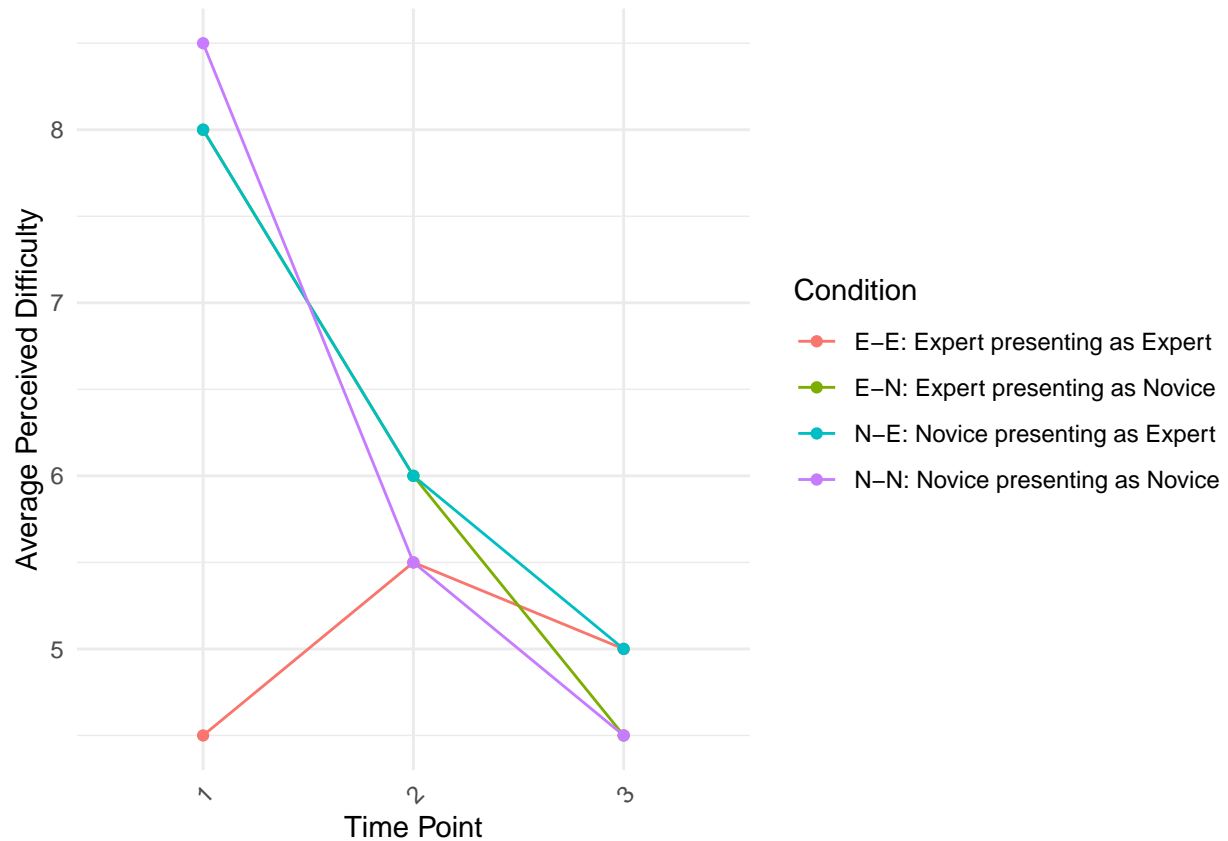
```
# confidence plot
plot_confidence <- ggplot(df_summary %>% filter(metric == "confidence"), aes(x = time, y = mean, color = condition)) +
  geom_line() +
  geom_point() +
  # geom_errorbar(aes(ymin = mean - se, ymax = mean + se), width = 0.2) + ## nicer to have when condition is not in facet
  labs(x = "Time Point", y = "Average Confidence", color = "Condition") +
  theme_minimal() +
  # facet_wrap(~condition) + # to have 4 separate graphs
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

plot_confidence
```



```
# difficulty plot
plot_difficulty <- ggplot(df_summary %>% filter(metric == "difficulty"), aes(x = time, y = mean, color = Condition)) +
  geom_line() +
  geom_point() +
  # geom_errorbar(aes(ymin = mean - se, ymax = mean + se), width = 0.2) + ## nicer to have when condition is not expert
  labs(x = "Time Point", y = "Average Perceived Difficulty", color = "Condition") +
  theme_minimal() +
  # facet_wrap(~condition) + # to have 4 separate graphs
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

plot_difficulty
```



Friedman Test (Non-Parametric ANOVA)

```
# Friedman Test for Confidence
friedman_confidence <- friedman.test(
  score ~ time | id,
  data = df_long %>% filter(metric == "confidence")
)

# Friedman Test for Difficulty
friedman_difficulty <- friedman.test(
  score ~ time | id,
  data = df_long %>% filter(metric == "difficulty")
)

# Results!
#friedman_confidence
#friedman_difficulty

cat("\nFriedman Test Results for Confidence:\n")
```


Friedman Test Results for Confidence:

```
cat("Chi-squared: ", friedman_confidence$statistic, "\n")
```

Chi-squared: 7.357143

```

cat("Degrees of Freedom: ", friedman_confidence$parameter, "\n")

## Degrees of Freedom: 2

cat("p-value: ", friedman_confidence$p.value, "\n")

## p-value: 0.02525903

if (friedman_confidence$p.value < 0.05) {
  cat("Interpretation: There is a significant difference in confidence across the time points.\n")
} else {
  cat("Interpretation: There is no significant difference in confidence across the time points.\n")
}

## Interpretation: There is a significant difference in confidence across the time points.

cat("\nFriedman Test Results for Difficulty:\n")

##
## Friedman Test Results for Difficulty:

cat("Chi-squared: ", friedman_difficulty$statistic, "\n")

## Chi-squared: 6.689655

cat("Degrees of Freedom: ", friedman_difficulty$parameter, "\n")

## Degrees of Freedom: 2

cat("p-value: ", friedman_difficulty$p.value, "\n")

## p-value: 0.0352663

if (friedman_difficulty$p.value < 0.05) {
  cat("Interpretation: There is a significant difference in difficulty across the time points.\n")
} else {
  cat("Interpretation: There is no significant difference in difficulty across the time points.\n")
}

## Interpretation: There is a significant difference in difficulty across the time points.

```

Post-Hoc Testing

```

df_long$score <- as.numeric(df_long$score)
df_long$time <- as.factor(df_long$time)

## NOT WORKING AT THE MOMENT

# Post-Hoc Wilcoxon Signed-Rank Test for Confidence (Pairwise comparisons)
pairwise_wilcox_confidence <- df_long %>%
  filter(metric == "confidence") %>%
  pairwise.wilcox.test(score ~ time, paired = TRUE, p.adjust.method = "bonferroni")

str(df_long)

# Post-Hoc Wilcoxon Signed-Rank Test for Difficulty (Pairwise comparisons)
pairwise_wilcox_difficulty <- df_long %>%
  filter(metric == "difficulty") %>%
  pairwise.wilcox.test(score, time, paired = TRUE, p.adjust.method = "bonferroni")

# Results!

cat("\nPost-Hoc Wilcoxon Signed-Rank Test Results for Confidence:\n")
print(pairwise_wilcox_confidence)

cat("\nPost-Hoc Wilcoxon Signed-Rank Test Results for Difficulty:\n")
print(pairwise_wilcox_difficulty)

```