# Experiment Analysis

Maya Elasmar

2025-01-20

## Introduction

This document gives an overview of the analysis of the measures and open questions used for evaluating the integration of a pedagogical agent in counselling simulation training for child helplines. The document includes analysis of:

- T-test with deltas Analysis of the performance and self-efficacy: The performance calculation and the self-efficacy questionnaire are available in the thesis PDF, chapter 4 and appendix B.5, respectively.
- T-test Analysis of the perceived usefulness: The questionnaire is available in the thesis PDF, appendix B.6.
- Analysis of the Double coding of a thematic analysis for the open questions. There were 2 questions: 1. "What did you like the most about our system?" 2. "What did you like the least about our system?"

We need the following files for these analyses:

- performance_data_control_pre.csv: The averages of performance of each participant of the control group in the pre-session.
- performance_data_control_post.csv: The averages of performance of each participant of the control group in the post-session.
- performance_data_intervention_pre.csv: The averages of performance of each participant of the intervention group in the pre-session.
- performance_data_intervention_post.csv: The averages of performance of each participant of the intervention group in the post-session.
- average_SE_control_pre.csv: The averages of self-efficacy scores of each participant of the control group in the pre-session.
- average_SE_control_post.csv: The averages of self-efficacy scores of each participant of the control group in the post-session.
- average_SE_intervention_pre.csv: The averages of self-efficacy scores of each participant of the intervention group in the pre-session.
- average_SE_intervention_post.csv: The averages of self-efficacy scores of each participant of the intervention group in the post-session.
- averages_PU_control.csv: The averages of perceived usefuleness scores of each participant of the control group.

- averages_PU_intervention.csv: The averages of perceived usefuleness scores of each participant of the intervention group.
- Responses Coders Assignment.csv: The coders themes assignment to the the respones of the open questions.

These files are the final results after data cleaning in the python files: Experiment_Data.ipynb , Control_Performance.ipynb and Intervention_Performance.ipynb.

This document includes also the original power analysis that we did to calculate the sample size. In addition, it includes the projected power analysis that we did after the experiment now we have calculated the correlation and effect size.

```
library(readr) # Data reading
library(effsize) # Effect Size
```

```
## Warning: package 'effsize' was built under R version 4.3.3
```

### Measures Data Reading

```
# Read Performance Data
inter_performance_pre <- read_csv("performance_data_intervention_pre.csv")
inter_performance_post <- read_csv("performance_data_intervention_post.csv")

control_performance_pre <- read_csv("performance_data_control_pre.csv")
control_performance_post <- read_csv("performance_data_control_post.csv")

# Read Self-Efficacy Data
inter_SE_pre <- read_csv("averages_SE_intervention_pre.csv")
inter_SE_post <- read_csv("averages_SE_intervention_post.csv")

control_SE_pre <- read_csv("averages_SE_control_pre.csv")
control_SE_post <- read_csv("averages_SE_control_post.csv")

# Read Perceived Usefulness Data
inter_PU <- read_csv("averages_PU_intervention.csv")
control_PU <- read_csv("averages_PU_control.csv")
```

### Calculating Deltas

```
# Calculate Deltas of Performance
inter_performance_deltas <- inter_performance_post$`Performance Data` - inter_performance_pre$`Performan
control_performance_deltas <- control_performance_post$`Performance Data` - control_performance_pre$`Per

# Calculate Deltas of Self-Efficacy
inter_SE_deltas <- inter_SE_post$`P_Average` - inter_SE_pre$`P_Average`
control_SE_deltas <- control_SE_post$`P_Average` - control_SE_pre$`P_Average`
```

### T-test Analysis

```
# Perform Independent t-test on the Deltas
t_test_result_performance <- t.test(inter_performance_deltas, control_performance_deltas, paired = FALSE)
t_test_result_SE<- t.test(inter_SE_deltas, control_SE_deltas, paired = FALSE)

# Perform Independent t-test
t_test_result_PU<- t.test(inter_PU$`P_Average`, control_PU$`P_Average`, paired = FALSE)
```

### Print the t-test Results

```
# Print the t-test result of performance
print(t_test_result_performance)
```

```
##
##  Welch Two Sample t-test
##
## data:  inter_performance_deltas and control_performance_deltas
## t = 1.4607, df = 19.989, p-value = 0.1596
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.05126063  0.29073581
## sample estimates:
##  mean of x  mean of y
## 0.12775898 0.00802139
```

```
# Print the t-test result of Self-Efficacy
print(t_test_result_SE)
```

```
##
##  Welch Two Sample t-test
##
## data:  inter_SE_deltas and control_SE_deltas
## t = 0.73392, df = 16.687, p-value = 0.4732
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.7021903  1.4496651
## sample estimates:
##   mean of x   mean of y
##  0.04040404 -0.33333333
```

```
# Print the t-test result of Perceived Usefulness
print(t_test_result_PU)
```

```
##
##  Welch Two Sample t-test
##
## data:  inter_PU$P_Average and control_PU$P_Average
## t = 0.49078, df = 19.92, p-value = 0.6289
```

```
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.5841334  0.9434408
## sample estimates:
## mean of x mean of y
##  5.467532  5.287879
```

### Calculate The Standard Deviation (SD)

```
# The SD calculation of Performance
sd_inter_performance_deltas <- sd(inter_performance_deltas)
sd_control_performance_deltas <- sd(control_performance_deltas)

# The SD calculation of Self-Efficacy
sd_inter_SE_deltas <- sd(inter_SE_deltas)
sd_control_SE_deltas <- sd(control_SE_deltas)

# The SD calculation of Perceived Usefulness
sd_inter_PU <- sd(inter_PU$`P_Average`)
sd_control_PU <- sd(control_PU$`P_Average`)
```

### Print The SDs

```
# Print SDs of Performance
print(paste("SD Intervention Performance Deltas:", sd_inter_performance_deltas))
```

```
## [1] "SD Intervention Performance Deltas: 0.189933040521482"
```

```
print(paste("SD Control Performance Deltas:", sd_control_performance_deltas))
```

```
## [1] "SD Control Performance Deltas: 0.194525377984694"
```

```
# Print SDs of Self-Efficacy
print(paste("SD Intervention SE Deltas:", sd_inter_SE_deltas))
```

```
## [1] "SD Intervention SE Deltas: 0.889267596094406"
```

```
print(paste("SD Control SE Deltas:", sd_control_SE_deltas))
```

```
## [1] "SD Control SE Deltas: 1.43587199814668"
```

```
# Print SDs of Perceived Usefulness
print(paste("SD Intervention PU:", sd_inter_PU))
```

```
## [1] "SD Intervention PU: 0.830762887881531"
```

```
print(paste("SD Control PU:", sd_control_PU))
```

```
## [1] "SD Control PU: 0.885346476718795"
```

### Calculate The Correlation

```
# Calculate Correlation for Performance Data
corr_inter_perf <- cor(inter_performance_pre$`Performance Data`, inter_performance_post$`Performance Da
print(paste("Correlation Intervention Performance: ", corr_inter_perf))
```

```
## [1] "Correlation Intervention Performance:  -0.35811154109131"
```

```
corr_control_perf <- cor(control_performance_pre$`Performance Data`, control_performance_post$`Performa
print(paste("Correlation Control Performance: ", corr_control_perf))
```

```
## [1] "Correlation Control Performance:  0.259923570194931"
```

```
# Calculate Correlation for Self-Efficacy Data
corr_inter_se <- cor(inter_SE_pre$`P_Average`, inter_SE_post$`P_Average`)
print(paste("Correlation Intervention SE: ", corr_inter_se))
```

```
## [1] "Correlation Intervention SE:  0.543347676035446"
```

```
corr_control_se <- cor(control_SE_pre$`P_Average`, control_SE_post$`P_Average`)
print(paste("Correlation Control SE: ", corr_control_se))
```

```
## [1] "Correlation Control SE:  0.296913682072512"
```

### Calculate the Effect Size

```
# Calculate the effect size for Performance
cohen_d_performance <- cohen.d(inter_performance_deltas, control_performance_deltas)
print(cohen_d_performance)
```

```
##
## Cohen's d
##
## d estimate: 0.6228452 (medium)
## 95 percent confidence interval:
##      lower      upper
## -0.2879231  1.5336136
```

```
# Calculate the effect size for Self-Efficacy
cohen_d_SE <- cohen.d(inter_SE_deltas, control_SE_deltas, pooled = TRUE)
print(cohen_d_SE)
```

```
##
## Cohen's d
##
## d estimate: 0.3129441 (small)
## 95 percent confidence interval:
##      lower      upper
## -0.5819414  1.2078296
```

```
#Calculate the effect size for Perceived Usefulness
cohen_d_PU <- cohen.d(inter_PU$`P_Average`, control_PU$`P_Average`, pooled = TRUE)
print(cohen_d_PU)
```

```
##
## Cohen's d
##
## d estimate: 0.2092674 (small)
## 95 percent confidence interval:
##      lower      upper
## -0.6826216  1.1011564
```

## Thematic Analysis

```
library(psych)   # multivariate analysis
```

```
##
## Attaching package: 'psych'

## The following object is masked from 'package:effsize':
##
##     cohen.d
```

### Read the double Coding Data

```
coding_data <- read.csv("Responses Coders Assignment.csv")
```

### Calculate the confusion matrix

```
confusion_matrix <- table(coding_data$Coder.1, coding_data$Coder.2)

# Print the confusion matrix
print(confusion_matrix)
```

```
##
##                       Ease of use Guidance Lack of Control Lack of Guidance
##   Ease of use                   4        0               0                0
##   Guidance                      3        4               0                0
```

```
##   Lack of Control              0        0               2                1
##   Lack of Guidance             0        0               0                1
##   Miscommunication             0        0               3                1
##   Nan                          0        0               0                0
##   Realism                      0        0               0                0
##   Skill Development            0        0               0                0
##
##                     Miscommunication Nan Realism Skill Development
##   Ease of use                      0   0       0                 0
##   Guidance                         0   0       0                 2
##   Lack of Control                  0   0       0                 0
##   Lack of Guidance                 0   0       0                 0
##   Miscommunication                13   0       0                 0
##   Nan                              0   1       0                 0
##   Realism                          0   0       2                 0
##   Skill Development                0   0       0                 7
```

### Calculate the Cohen's Kappa

```
# Calculate Cohen's Kappa
kappa_result <- cohen.kappa(confusion_matrix)

# Display the result
print(kappa_result)
```

```
## Call: cohen.kappa1(x = x, w = w, n.obs = n.obs, alpha = alpha, levels = levels,
##     w.exp = w.exp)
##
## Cohen Kappa and Weighted Kappa correlation coefficients and confidence boundaries
##                  lower estimate upper
## unweighted kappa  0.57     0.72  0.87
## weighted kappa    0.60     0.81  1.00
##
##  Number of subjects = 44
```

## Original Power Analysis

### Set Constants

```
set.seed(123)

# Constants
n_participants <- 11  # Number of participants
n_simulations <- 1000  # Number of simulations
correlation <- 0.71  # Correlation between baseline and post
R_squared <- correlation^2  # Calculate R-square
effect_size <- 0.8  # Effect size for Condition B
```

### Function to Generate Results

```r
generate_condition <- function(n_participants, n_simulations, R_squared, effect_size = 0) {
  # Baseline
  baseline <- matrix(rnorm(n_participants * n_simulations, mean = 0, sd = 1),
                     nrow = n_simulations, ncol = n_participants)

  # Post
  post <- baseline * R_squared +
    matrix(rnorm(n_participants * n_simulations, mean = 0, sd = 1),
           nrow = n_simulations, ncol = n_participants) * (1 - R_squared)

  # Apply effect size adjustment for post if provided
  post <- post + effect_size

  # Delta (Post - Baseline)
  delta <- post - baseline

  return(list(baseline = baseline, post = post, delta = delta))
}
```

### Generate Results

```r
# Generate results for Condition A
condition_A <- generate_condition(n_participants, n_simulations, R_squared)

# Generate results for Condition B (with effect size adjustment)
condition_B <- generate_condition(n_participants, n_simulations, R_squared, effect_size)
```

### Calculate the Deltas

```r
mean_delta_A <- apply(condition_A$delta, 1, mean)
mean_delta_B <- apply(condition_B$delta, 1, mean)
```

### Calculta The t-test and Power

```r
# Constants
alpha <- 0.05  # Significance level

# Perform t-tests for each simulation
p_values <- sapply(1:n_simulations, function(i) {
  t_test <- t.test(condition_A$delta[i, ], condition_B$delta[i, ])
  return(t_test$p.value)
})

# Count p-values less than alpha
significant_tests <- sum(p_values < alpha)
```

```r
# Calculate power
power <- significant_tests / n_simulations

# Output results
cat("Number of significant tests (p < 0.05):", significant_tests, "\n")
```

```
## Number of significant tests (p < 0.05): 718
```

```r
cat("Statistical Power:", power, "\n")
```

```
## Statistical Power: 0.718
```

### Projected Power Analysis

For this we change only the correlation and the effect size to the highest calculated average correlation and effect size of the measurements.

```r
# Constants
n_participants <- 54# Number of participants
n_simulations <- 1000  # Number of simulations
correlation <- -0.36  # Correlation between baseline and post
R_squared <- correlation^2  # Calculate R-square
effect_size <- 0.62  # Effect size
```

After this you follow, the rest of the rest of the original power analysis, which should result in a power of approx. 0.72.