

Statistical Analysis

Evaluation of a BDI-based Virtual Agent for Training Child Helpline Counsellors

Sharon Afua Grundmann, Mohammed Al Owayyed

August 2023

Introduction

This document gives an overview of the analysis of the questionnaires used for evaluating the BDI-based conversational agent in a paper “Lilobot: a cognitive conversational agent to train child helpline counselors on social support”. The document include analysis of:

- Analysis of the Double coding of a thematic analysis used for evaluating the BDI-based conversational agent. There are three questions: 1) “What was the best thing about your experience using Lilobot?”, 2) “What was the worst thing about your experience using Lilobot?”, and 3) “What do you think of the feedback you received at the end of your conversation with Lilobot?”.
- Counselling Self_efficacy Five Phase Model Questionnaire: measures participants’ Self_efficacy towards carrying out tasks related to the Five Phase Model (FPM).
- Perceived Influence on Learning Outcome (PILO) (i.e., perceived usefulness): measures participants’ perceived influence of the conversational agent on their knowledge of the Five Phase Model, their attitude towards conversational agents and their Self_efficacy through self assessment.
- System Usability Scale (SUS): measures participants’ subjective assessments of usability of the conversational agent.
- BDI outcomes: comparing the BDI model outcomes in the first sessions with the third session to see if it increased.
- Thematic analysis: assigning themes to the participants comments on the qualitative questions.

We need five files in total that corresponde to the measured variables:

- Self_efficacy_scores.csv: contains the counselling Self_efficacy scores measured pre- and post- the training interventions. This is based on the Counselling Self_efficacy Five Phase Model Questionnaire.
- useful_usab_scores.csv: contains the PILO and SUS scores of the participants.
- BDI_scores.csv: contains the BDI outcomes of the participants during the first and the third training sessions with the conversational agent.
- themes_data: participants quotes and their themes for the qualitative questions
- inter-reliability.csv: the double coding for the three open ended questions

Libraries

```

library(psych)      # multivariate analysis
library(dplyr)      # data manipulation
library(ggplot2)    # plotting graphs
library(ggpubr)     # plotting graphs
library(tidyverse)  # data manipulation and visualization
library(rstatix)    # pipe-friendly R functions
library(lsmeans)    # linear models
library(multcomp)   # hypothesis testing
# library(plyr)
library(pander)     # markdown
library(readxl)
library(nlme)
library(dplyr)
library(irr)

```

Reading Data

```

SEall = read.csv("Self-efficacy_scores.csv", sep = ";")

useful_usab_scores = read.csv("useful_usab_scores.csv", sep = ";")

BDI_scores = read.csv("BDI_scores.csv", sep = ";")

coding = read.csv("inter-reliability.csv", sep = ";")

themes = read.csv("themes_data.csv", sep = ";")

```

Mean of counseling experience

```

#how many participants do we have?
nrow(SEall)

```

```
## [1] 28
```

```

# get data from the file
exp <- SEall[c(27)]

```

```

#calculating the mean and standard deviation of the years of experience
exp <- rowMeans(exp) # make it a list
exp <- na.omit(exp) # remove empty spaces
mean(exp) #get the mean of years of experience

```

```
## [1] 3.537037
```

```
sd(exp) #standard deviation
```

```
## [1] 3.949016
```

for the 28 counselors, The years of experience for counselors ($M = 3.54$ years, $SD = 3.95$).

cleaning the Self_efficacy data and removing empty questionnaires

we will start by cleaning and preparing the data:

```
# check if an entire questionnaire is missing by a participant
pre <- SEall[c(2:9)] # all pre questionnaires in one dataset
pre <- mutate_all(pre, function(x) as.numeric(as.character(x))) #change data to numerical

# all post questionnaires in the first condition (groups A&B) in one dataset
post1 <- SEall[c(10:17)]
post1 <- mutate_all(post1, function(x) as.numeric(as.character(x))) #change to numerical

# all post questionnaires in the second condition (groups A&B) in one dataset
post2 <- SEall[c(18:25)]
post2 <- mutate_all(post2, function(x) as.numeric(as.character(x))) #change to numerical
# find missing values
sum(is.na(pre)) # no missing value
```

```
## [1] 0
```

```
sum(is.na(post1)) # 8 missing values
```

```
## [1] 8
```

```
post1 # looking at the data, the 8 values are from one participant in row 5
```

```
##      Q111_1 Q112_1 Q113_1 Q114_1 Q115_1 Q116_1 Q117_1 Q118_1
## 1         2      -2         1         2         0         2      -3         2
## 2         5         2         2        -2         2         5         3         3
## 3         5         4         4         4         4         5         5         5
## 4         5         5         5         4         5         5         5         5
## 5        NA        NA        NA        NA        NA        NA        NA        NA
## 6         5         5         5         5         5         5         5         2
## 7         5         4         4         4         4         5         4         4
## 8         5         5         5         5         5         5         5         5
## 9         4         3         3         3         1         3         3         3
## 10        3         3         3         3         4         3         3         4
## 11        3         3         2         2         3         5         4         4
## 12        4         3         4         4         2         4         4         3
## 13        3         1         1         1         1         3         4        -1
## 14        5         5         5         5         5         5         5         5
## 15        5         3         2         2         2         4         5         5
## 16        5         5         5         5         5         5         5         5
## 17       -2        -5        -5        -2        -5        -5        -2        -5
## 18        4         5         3         0         2         4         3         3
## 19        5         5         5         2         2         1         1         1
## 20        4         4         4         3         4         5         5         4
## 21        5         4         4         4         4         4         4         4
## 22        2         2         3         2         3         2         3         3
## 23        3         2         3        -1         2         4         3        -1
## 24        3         2         2         3         2         2         5         4
## 25        5         5         4         4         4         4         5         5
```

```
## 26      5      5      4      4      4      4      5      5
## 27      3      3      3      2      2      3      3      1
## 28      3      3      3      3      3      3      3      3
```

```
sum(is.na(post2)) # 11 missing values from six participants
```

```
## [1] 11
```

```
post2 #Looking at the data, we can take the average for participants missing the values
```

```
##      Q120_1 Q121_1 Q122_1 Q123_1 Q124_1 Q125_1 Q126_1 Q127_1
## 1         1      -1      -2        2        1        4       -3        3
## 2         5        5        0       -3        0        5        0        2
## 3         5        5       -1        3        1        5        5       NA
## 4         5        4        4       -4       -4        4        4        5
## 5         4       -2        1       -4       -3        3        3       NA
## 6         5       -2       -3       -3       -5       -5        1       -5
## 7         5        4        4        4        5        5        4        4
## 8         5        0        0        0       -5       -5        0        0
## 9         3       NA       NA       -2       -5       -5       NA       NA
## 10        3        3        3        3        3        3        3        4
## 11        4        2        2       -3        1       -4        1       -2
## 12        2        0        1        1       -1       NA        2       -1
## 13        1        1       -2       -2       -3        2        1        1
## 14        5        5        5        5        5        5        5        5
## 15        2        2       -3       -2       -4       NA        1        1
## 16        5        3       -2       -2       -5        5        5        5
## 17       -2       -2       -5       -5       -5       -5        2       -5
## 18       -5       -5       -5       -5       -5       -5       -5       -5
## 19        5        5        3       -2        2        2        5       -1
## 20        5        4        4        3        4        4        5        4
## 21        3        2       NA       NA       NA        3        2        2
## 22        1        2       -1       -1        0       -1        1       -1
## 23        4       -3       -1        1       -3        2        5       -1
## 24        1        1        1        1        1        1        1        1
## 25        5        3        3        3        4        4        5        5
## 26        5        3        3        3        4        4        5        5
## 27        3        3        3        2        1        4        3        2
## 28        3        2        2        0       -2       -2        2       -2
```

there are 7 people with missing values:

for post1 measures, 8 questions are missing from one person who didn't answer any question from the questionnaire, this will affect the analysis of both the chatbot condition and the text condition. i.e., for participants 5, we have pre-chatbot but not post-chatbot, and we have post-text but not pre-text. Therefore, including their data is not possible and we decided to remove this row instead (only for Self_efficacy).

As for post2, the 11 missing values are from six different participants who filled at least one question in the questionnaire. So, we will take the average of the rest of the values

Double coding for thematic analysis

set up variables

```
# the coding for the first question (includes coder 1 and 2)
question1 <- coding %>% dplyr::select(coder1nq1, coder2nq1)
# the coding for the second question (includes coder 1 and 2)
question2 <- coding %>% dplyr::select(coder1nq2, coder2nq2)
# the coding for the third question (includes coder 1 and 2)
question3 <- coding %>% dplyr::select(coder1nq3, coder2nq3)
```

Calculate Cohen's kappa for the three open ended questions and print them

```
kappa1 <- kappa2(question1) #inter-reliability agreement of question 1
kappa2 <- kappa2(question2) #inter-reliability agreement of question 2
kappa3 <- kappa2(question3) #inter-reliability agreement of question 3
```

```
print(kappa1)
```

```
## Cohen's Kappa for 2 Raters (Weights: unweighted)
##
## Subjects = 30
## Raters = 2
## Kappa = 0.63
##
## z = 7.56
## p-value = 4.06e-14
```

```
print(kappa2)
```

```
## Cohen's Kappa for 2 Raters (Weights: unweighted)
##
## Subjects = 32
## Raters = 2
## Kappa = 0.522
##
## z = 5.81
## p-value = 6.3e-09
```

```
print(kappa3)
```

```
## Cohen's Kappa for 2 Raters (Weights: unweighted)
##
## Subjects = 28
## Raters = 2
## Kappa = 0.677
##
## z = 7.29
## p-value = 3.06e-13
```

The inter-reliability resulted in a substantial agreement for the first (Cohen's $\kappa = 0.63$) and third (Cohen's $\kappa = 0.68$) questions, and a moderate agreement for the second (Cohen's $\kappa = 0.52$).

Self_efficacy tests

Here, we will test the self efficacy measures in both conditions: the chatbot and the text based. First, we will set up the dataset:

```
# get data
a <- subset(SEall, Group == "A") # Group (A) is the group who did chatbot (post1) then text-based condi
b <- subset(SEall, Group == "B") # Group (B) is the group who did text-based (post1) then chatbot condi

a <- a[-2,] # the person with the missing data in the post Self_efficacy is dropped

a_pre <- a[c(2:9)]           #pre measures for group a
a_post1 <- a[c(10:17)]       #first condition (chatbot) post measures for group a
a_post2 <- a[c(18:25)]       #second condition (text) post measures for group a

b_pre <- b[c(2:9)]           #pre measures for group b
b_post1 <- b[c(10:17)]       #first condition (text) post measures for group b
b_post2 <- b[c(18:25)]       #second condition (chatbot) post measures for group b
```

Now, we will devide the data into pre-chatbot, post-chatbot, and pre-text, post-text. Then, we will assign the new dataset.

```
# compute Self_efficacy scores based on only present values. the average of participants
#with empty values will only be taken for the available data.

#for chatbot
chatbot_pre = c(rowMeans(a_pre, na.rm=TRUE), rowMeans(b_post1, na.rm=TRUE))
chatbot_post = c(rowMeans(a_post1, na.rm=TRUE), rowMeans(b_post2, na.rm=TRUE))

#for text
text_pre = c(rowMeans(a_post2, na.rm=TRUE), rowMeans(b_pre, na.rm=TRUE))
text_post = c(rowMeans(a_post2, na.rm=TRUE), rowMeans(b_post1, na.rm=TRUE))

# combine datasets. Self_efficacy now has the user id, the training type, the pre measure value
#and the post measure value

Self_efficacy = data.frame(id = seq.int(27), training = rep(c("chatbot", "text"), each = 27),
  pre = c(chatbot_pre, text_pre), post = c(chatbot_post, text_post))

# change the data structure of the columns (pre, post) to be on a single column (time) for ANOVA test

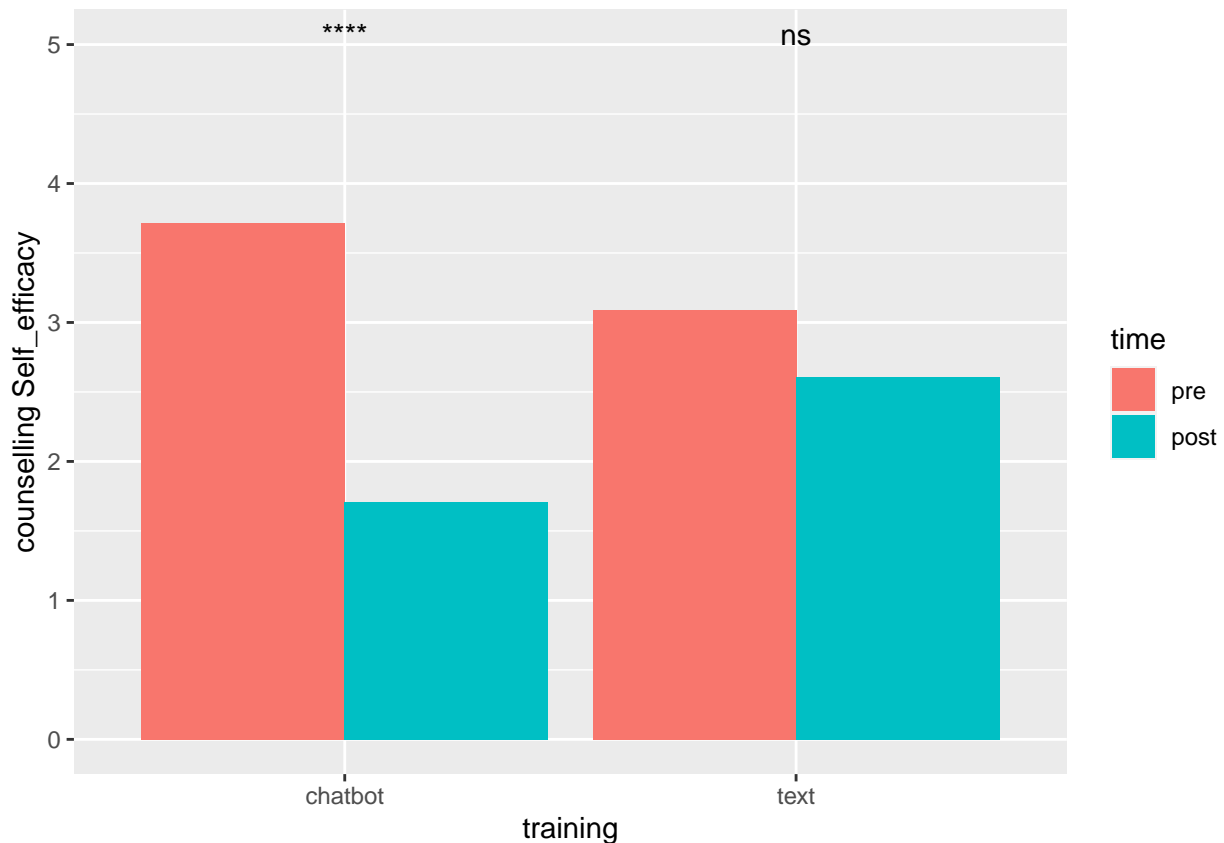
#transfer the dataset to Self_efficacy.long columns to be: the participants id, the training type
 #(chatbot or text), time of measurement (pre and post), the value of the pre or post

Self_efficacy.long <- Self_efficacy %>%
  gather(key = "time", value = "self_efficacy", pre, post)
```

```
Self_efficacy.long$time = factor(Self_efficacy.long$time , levels=c("pre", "post"))
Self_efficacy.long$training = factor(Self_efficacy.long$training , levels=c("chatbot", "text"))
```

Let's visualise the means for the chatbot and the text conditions based on time of measurement

```
# barplot to compare pre & post of the conditions.
bar <- ggplot(Self_efficacy.long, aes(training , self_efficacy, fill = time), ylab="counselling Self_ef.
  xlab("training") + ylab("counselling Self_efficacy")
bar + stat_summary(fun = mean, geom = "bar", position="dodge") +
  stat_compare_means(label="p.signif", method="wilcox.test", paired = TRUE)
```



As the data is ready to be tested, we will fit a model to check the overall effect.

```
# anova
#fitting a mixed effects model for the repeated measures. To check the overall
#effect on training, time and their interaction effect
model <- lme(self_efficacy ~ training * time, data = Self_efficacy.long, random = ~ 1|id)
pander(anova(model),
caption = "Effect of training, time and interaction effect on counselling Self_efficacy")
```

Table 1: Effect of training, time and interaction effect on counselling Self_efficacy

	numDF	denDF	F-value	p-value
(Intercept)	1	78	104.9	4.441e-16
training	1	78	0.2036	0.6531
time	1	78	17.32	8.076e-05
training:time	1	78	6.521	0.01261

```
#Get the mean and SD of post and pre measures
```

```
mean(Self_efficacy$post) #mean of the post measures (regardless of the condition)
```

```
## [1] 2.156548
```

```
SD(Self_efficacy$post) #standard deviation of the post measures (regardless of the condition)
```

```
## [1] 2.385616
```

```
mean(Self_efficacy$pre) #mean of the pre measures (regardless of the condition)
```

```
## [1] 3.402778
```

```
SD(Self_efficacy$pre) #standard deviation of the pre measures (regardless of the condition)
```

```
## [1] 1.442831
```

Looking at the result, we can see that only a significant difference was found in the contrast Chatbot, and not in contrast Text.

We report the results as follows. A linear model was fitted on the counselling Self_efficacy of participants, taking the training intervention and time of measurement as independent variables, and including a two-way interaction between these variables. The analysis revealed no significant main effect on counseling Self_efficacy based on the type of intervention ($F(1, 78) = 0.2$, $p = .65$). However, we observed a significant main effect at different times of measurement ($F(1, 78) = 17.32$, $p < .001$), where post-counseling Self_efficacy ($M = 2.16$, $SD = 2.39$) was significantly lower than pre-counseling Self_efficacy ($M = 3.4$, $SD = 1.44$). The analysis also found a significant two-way interaction effect ($F(1, 78) = 6.52$, $p < .05$) between these two variables.

We will examine each condition further through

```
#simple effect analysis with interventions - chatbot, text
```

```
# merge the two factors (time and training). the new column name is timeANDtraining
```

```
Self_efficacy.long$timeANDtraining <- interaction(Self_efficacy.long$time, Self_efficacy.long$training)
```

```
levels(Self_efficacy.long$timeANDtraining)
```

```
## [1] "pre.chatbot" "post.chatbot" "pre.text" "post.text"
```

```

contrastChatbot <- c(1, -1, 0, 0) # to define the levels of pre & post for the chatbot
contrastText <- c(0, 0, 1, -1) # to define the levels of pre & post for the text

simpleEff <- cbind(contrastChatbot, contrastText)
contrasts(Self_efficacy.long$timeANDtraining) <- simpleEff # the levels are defined

#fitting a model to check the interaction effect on each condition
simpleEffectModel <-lme(self_efficacy ~ timeANDtraining , random = ~1|id,
                        data = Self_efficacy.long, na.action = na.exclude)
pander(simpleEffectModel)

```

Table 2: Linear mixed-effects model fit by REML : self_efficacy ~ timeANDtraining To examine the two-way interaction further, we used a simple effect analysis. This analysis revealed a significant difference ($t(78) = 4.75$, $p < .001$) in counselling Self_efficacy before and after training for the chatbot intervention, but no significant effect ($t(78) = 1.14$, $p = .26$) was found in the text intervention across the two-time points of measurement

	Value	Std.Error	DF	t-value	p-value
(Intercept)	2.78	0.2714	78	10.24	4.318e-16
timeANDtrainingcontrastChatbot	1.005	0.2118	78	4.748	9.155e-06
timeANDtrainingcontrastText	0.2407	0.2118	78	1.137	0.2591
timeANDtraining	0.1351	0.2995	78	0.4512	0.6531

Perceived usefulness (PILO)

From the PILO questionnaire, we will check the deviation from the neutral zero scores for each question. This questionnaire was a post measure only for the chatbot condition.

```

# pilo analysis

# get data from the csv
pilo <- useful_usab_scores[c(2:9)]
pilo <- mutate_all(pilo, function(x) as.numeric(as.character(x))) # change to numeric

```

```

# wilcoxon-test on each question
wil<- apply(pilo,2,wilcox.test, mu = 0)
print(wil) # only Q4_1 and Q7_1 have a significant p-value

```

```

## $Q3_1
##
## Wilcoxon signed rank test with continuity correction
##
## data:  newX[, i]
## V = 14, p-value = 1
## alternative hypothesis: true location is not equal to 0
##
##

```

```

## $Q4_1
##
## Wilcoxon signed rank test with continuity correction
##
## data: newX[, i]
## V = 17.5, p-value = 0.5877
## alternative hypothesis: true location is not equal to 0
##
##
## $Q5_1
##
## Wilcoxon signed rank test with continuity correction
##
## data: newX[, i]
## V = 32, p-value = 0.6795
## alternative hypothesis: true location is not equal to 0
##
##
## $Q6_1
##
## Wilcoxon signed rank test with continuity correction
##
## data: newX[, i]
## V = 14, p-value = 0.5201
## alternative hypothesis: true location is not equal to 0
##
##
## $Q7_1
##
## Wilcoxon signed rank test with continuity correction
##
## data: newX[, i]
## V = 13.5, p-value = 0.02377
## alternative hypothesis: true location is not equal to 0
##
##
## $Q8_1
##
## Wilcoxon signed rank test with continuity correction
##
## data: newX[, i]
## V = 65, p-value = 0.1318
## alternative hypothesis: true location is not equal to 0
##
##
## $Q9_1
##
## Wilcoxon signed rank test with continuity correction
##
## data: newX[, i]
## V = 84, p-value = 0.965
## alternative hypothesis: true location is not equal to 0
##
##

```

```
## $Q10_1
##
## Wilcoxon signed rank test with continuity correction
##
## data: newX[, i]
## V = 32, p-value = 0.01101
## alternative hypothesis: true location is not equal to 0

#get mean and standard deviation of the 2 significant question
PiloSE <- na.omit(pilo$Q7_1) # remove empty spaces for the question about self-efficacy
PiloUse <- na.omit(pilo$Q10_1) # remove empty spaces for the question about self-efficacy
mean(PiloSE) #get the mean for the Self efficacy question

## [1] -1.058824

sd(PiloSE) #standard deviation for the Self efficacy question

## [1] 1.712841

mean(PiloUse) #get the mean for the Self efficacy question

## [1] -1.619048

sd(PiloUse) #standard deviation for the Self efficacy question

## [1] 2.558832

# get the z-value for the significant questions
qnorm(wil$Q7_1$p.value)

## [1] -1.981472

qnorm(wil$Q10_1$p.value)

## [1] -2.289934
```

We observe a significant deviation from the neutral zero point in two questions in the usefulness questionnaire. The first is about participants' Self_efficacy concerning the Five Phase Model, receiving a negative score ($Z = -1.98$, $p < .05$). The second is the participants' view regarding the usefulness of conversational agents as a learning tool, which also received a negative score ($Z = -2.29$, $p < .05$).

Usability (SUS)

Using SUS survey, we will rate the usability of the chatbot for the participants.

```

# sus analysis

# get data
sus <- useful_usab_scores[c(10:19)]
sus <- sus[-c(1), ] #remove question label row
sus <- mutate_all(sus, function(x) as.numeric(as.character(x))) #change to numerals

# transformation done based on SUS guidelines
odd <- function(x) x - 1 # for odd items, we need to subtract one from the user response of odd items
even <- function(x) 5 - x # for even items, we need to subtract the user responses from five

#applying the transformation to the items
temp <- data.frame(apply(sus[c(1, 3, 5, 7,9)],2, odd))
temp <- data.frame(temp, apply(sus[c(2,4,6,8,10)], 2, even))

# convert the mean to 0-100 by multiply by 2.5
susMean <- mean(rowSums(sus) * 2.5)
susMean #sus = 67.41071

## [1] 67.41071

# estimate standard deviation
susSd <- sd(rowSums(sus) * 2.5)
susSd # sd = 6.436375

```

```
## [1] 6.436375
```

We report an average score of 67.41 (SD = 6.44), which can be interpreted as ‘ok’

BDI outcomes

During the intervention with Lilobot, participants engaged with the agent in three consecutive sessions, each lasting approximately 15 minutes. The goal of the first and third sessions was to counsel Lilobot according to the Five Phase Model, while the second session allowed participants to explore the agent. We will test whether the BDI outcomes increased in the third session compared to the first.

```

# score session 3 is higher than session 1
# combine into one dataset
h4 <- data.frame(
  id=factor(1:28),
  session = rep(c("first", "third"), each = 28),
  outcome = c(BDI_scores$Session.1, BDI_scores$Session.3)
)

#transfer the data to conduct a t test
h4.long <- data.frame(
  id=factor(1:28),
  first = c(BDI_scores$Session.1),
  third = c(BDI_scores$Session.3)
)

```

```
# t-test on the first and the third session scores
res_h4 <- t.test(h4.long$first, h4.long$third, paired = TRUE)
res_h4 # p-value = 0.09753

##
## Paired t-test
##
## data: h4.long$first and h4.long$third
## t = -1.7214, df = 25, p-value = 0.09753
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -1.08300139 0.09684754
## sample estimates:
## mean of the differences
## -0.4930769
```

The p-value of the test is 0.09753, which is greater than the significance level $\alpha = 0.05$. We can conclude that the average BDI outcome in the first and third sessions are not significantly different.

```
# describe data and get the mean and SD of the first and third sessions
h4 %>%
  group_by(session) %>%
  get_summary_stats(outcome, type = "mean_sd")
```

```
## # A tibble: 2 x 5
##   session variable      n mean   sd
##   <chr>    <fct>    <dbl> <dbl> <dbl>
## 1 first    outcome      28  6.36  1.36
## 2 third    outcome      26  6.68  1.24
```

Qualitative analysis

First, we will assign the themes to the variable to know how many much time the themes were mentioned in the participants comments.

```
themesq1 <- themes[c(2:3)] # take the themes of the first question
themesq2 <- themes[c(5:6)] # take the themes of the second question
themesq3 <- themes[c(8:9)] # take the themes of the third question
themesq4 <- themes[c(10)] # take the answer for the fourth question
```

Let's start with the first question:

```
#merge the two themes columns into one
allthemesQ1 <- data.frame(a = c(themesq1$theme1q1, themesq1$theme2q1))
# print how many occurrence for each theme
table(allthemesQ1)
```

```
## allthemesQ1
##
Fast response Insight
```

```
##          37          6          2
##      Realistic      Reflective Self-directed learning
##          4          4          3
```

The second question:

```
#merge the two themes columns into one
allthemesQ2 <- data.frame(a = c(themesq2$theme1q2, themesq2$theme2q2))
# print how many occurrence for each theme
table(allthemesQ2)
```

```
## allthemesQ2
##          Backtracking          Emoticons
##          26          1          2
## Not conversing naturally      Repetitive answers      Segmentation
##          22          4          1
```

The third question:

```
#merge the two themes columns into one
allthemesQ3 <- data.frame(a = c(themesq3$theme1q3, themesq3$theme2q3))
# print how many occurrence for each theme
table(allthemesQ3)
```

```
## allthemesQ3
##          insightful no added value      No feedback      reflective
##          10          9          2          5          1
##          unclear
##          1
```

The fourth question:

```
# print how many occurrence for each theme
table(themesq4$RecommendLilo)
```

```
##
##  1  2  3
## 17  3  3
```

Each number refers to a different group (1: counselors-in-training, 2: novice counselors, 3: experienced counselors, 4: supervisors of the helpline)