Motivating individuals to complete thoughts-strengthening exercises via a conversational agent

Mohammed Al Owayyed

10/5/2020

Contents

1 Introduction

This document presents inferential statistical analyses of participants' perveived usefulness, self-efficacy, enjoyment, engagement, change in motivational barriers and change in thoughts believability. This analysis was reported in:

"Motivating PhD candidates with depression symptoms to complete thoughts-strengthening exercises via a conversational agent"

The OSF form belonging to this report can be found here: https://osf.io/v6tkq

Libraries used:

```r
library(foreign) #open various data files
library(tidyr)  # for wide to long format transformation of the data
library(ggplot2) # plotting & data
library(pander) # for pander tables
library(ez) #for ezANOVA
library(psych) # reliability function
library(stringr) #find how many repeated arugguments
library(pastecs) # plotting & data
library(lsr) # effect size
library(tidyverse) # visualize data
library(nlme) # for multilevel
library(lme4) # Non-linear multilevel
library(ggpubr) # plotting
library(rstatix) # for calculating effect size
library(psych) # reliability function
library(stringr) #find how many repeated arugguments
library(fitdistrplus) # to fit distribution
```

2 Data file

To read the data from the file:

```r
P_data <- read_excel('Dataset.xlsx', sheet = 'Sheet1')
```

Description of the data presented in P_data:

Field	Description
ParticipantID	The participant Unique ID
HighestBarrier	The highest rated motivational barrier by the participant
NotConvnced-Pre	The pre-input value for the barrier "Not Convinced" before doing the exercise
LowEff-Pre	The pre-input value for the barrier "Low Self-Efficacy" before doing the exercise
Perf-Pre	The pre-input value for the barrier "Perfectionism" before doing the exercise
NotConvnced-Post	The post-input value for the barrier "Not Convinced" before doing the exercise
LowEff-Post	The post-input value for the barrier "Low Self-Efficacy" before doing the exercise
Perf-Post	The post-input value for the barrier "Perfectionism" before doing the exercise
Group	Which condition the participant followed
Scenario	The scenario presented to rate the negative and positive thoughts from (a trip scenario, and an email scenario)

Field	Description
HighestNegThou	The highest rated negative thought by the participant
HighestPosThou	The highest rated positive by the participant
PreNegTh1	The pre value for the first negative thought
PreNegTh2	The pre value for the second negative thought
PreNegTh3	The pre value for the third negative thought
PrePosTh1	The pre value for the first positive thought
PrePosTh2	The pre value for the second positive thought
PrePosTh3	The pre value for the third positive thought
PostNegTh1	The post value for the first negative thought
PostNegTh2	The post value for the second negative thought
PostNegTh3	The post value for the third negative thought
PostPosTh1	The post value for the first positive thought
PostPosTh2	The post value for the second positive thought
PostPosTh3	The post value for the third positive thought
Chbot-Anecd	Did the participant see the social model strategy while dealing with the chatbot?
Chbot-Auth	Did the participant see the authority strategy while dealing with the chatbot?
Chbot-Arg	Did the participant see the argumentation strategy while dealing with the chatbot?
FirstChatPath	What is the first strategy presented by the chatbot
SecondChatPath	What is the second strategy presented by the chatbot
ThirdChatPath	What is the third strategy presented by the chatbot
ArgForNeg	The arguments written by participant which supports the negative thought
ArgAgainstNeg	The arguments written by participant which opposes the negative thought
ArgForPos	The arguments written by participant which supports the positive thought
ArgAgainstPos	The arguments written by participant which opposes the negative thought
Time-Spent	Time spent on doing the thought-strengthening exercise
Usefulness1	The post measure for the first usefulness question from 1 (strongly disagree) to 5 (strongly agree)
Usefulness2	The post measure for the second usefulness question from 1 (strongly disagree) to 5 (strongly agree)
Usefulness3	The post measure for the third usefulness question from 1 (strongly disagree) to 5 (strongly agree)
Usefulness4	The post measure for the fourth usefulness question from 1 (strongly disagree) to 5 (strongly agree)
Usefulness5	The post measure for the fifth usefulness question from 1 (strongly disagree) to 5 (strongly agree)
Usefulness6	The post measure for the sixth usefulness question from 1 (strongly disagree) to 5 (strongly agree)
Self-Efficacy1	The post measure for the first Self-Efficacy question from 0 (highly certain cannot do) to 10 (highly certain can do)
Self-Efficacy2	The post measure for the second Self-Efficacy question from 0 (highly certain cannot do) to 10 (highly certain can do)
Self-Efficacy3	The post measure for the third Self-Efficacy question from 0 (highly certain cannot do) to 10 (highly certain can do)
Self-Efficacy4	The post measure for the fourth Self-Efficacy question from 0 (highly certain cannot do) to 10 (highly certain can do)
Self-Efficacy5	The post measure for the fifth Self-Efficacy question from 0 (highly certain cannot do) to 10 (highly certain can do)
Compare	The enjoyment question post measure
Enjoy1	First question of the enjoyability survey from 1 (strongly disagree) to 5 (strongly agree)
Enjoy2	Second question of the enjoyability survey from 1 (strongly disagree) to 5 (strongly agree)

Field	Description
Enjoy3	Third question of the enjoyability survey from 1 (strongly disagree) to 5 (strongly agree)
Enjoy4	Fourth question of the enjoyability survey from 1 (strongly disagree) to 5 (strongly agree)
Enjoy5	Fifth question of the enjoyability survey from 1 (strongly disagree) to 5 (strongly agree)
Enjoy6	Sixth question of the enjoyability survey from 1 (strongly disagree) to 5 (strongly agree)
Enjoy7	Seventh question of the enjoyability survey from 1 (strongly disagree) to 5 (strongly agree)
Pre-Usefulness1	The pre measure for the first usefulness question from 1 (strongly disagree) to 5 (strongly agree)
Pre-Usefulness2	The pre measure for the second usefulness question from 1 (strongly disagree) to 5 (strongly agree)
Pre-Usefulness3	The pre measure for the third usefulness question from 1 (strongly disagree) to 5 (strongly agree)
Pre-Usefulness-4	The pre measure for the fourth usefulness question from 1 (strongly disagree) to 5 (strongly agree)
Pre-Usefulness5	The pre measure for the fifth usefulness question from 1 (strongly disagree) to 5 (strongly agree)
Pre-Usefulness6	The pre measure for the sixth usefulness question from 1 (strongly disagree) to 5 (strongly agree)
Pre-Self-Efficacy1	The pre measure for the first Self-Efficacy question from 0 (highly certain cannot do) to 10 (highly certain can do)
Pre-Self-Efficacy-2	The pre measure for the second Self-Efficacy question from 0 (highly certain cannot do) to 10 (highly certain can do)
Pre-Self-Efficacy-3	The pre measure for the third Self-Efficacy question from 0 (highly certain cannot do) to 10 (highly certain can do)
Pre-Self-Efficacy-4	The pre measure for the fourth Self-Efficacy question from 0 (highly certain cannot do) to 10 (highly certain can do)
Pre-Self-Efficacy-5	The pre measure for the fifth Self-Efficacy question from 0 (highly certain cannot do) to 10 (highly certain can do)
Pre-Compare	The enjoyment question pre measure

2.1 Missing data

the data of 29 participants who completed the exercise were excluded from the experiment and not included in the total 174 participants. The reasons for exclusion were performing the experiment more than once (n = 9), where only the first evaluation recording completed by the participants were included in the analysis; rating all pre-believability as 0, so it could not be lower in the post-believability measure (n = 5); writing nonsensical answers to the task questions (n = 13); and having the exact same answers as other participants to the open-ended questions (n = 2).

3 Perceived usefulness analysis

3.1 Reliability check

The participant in all three conditions were asked to fill a perceived usefulness questionnaire of before and after doing the exercise. The questionnaire includes 6 questions which they were asked to rate from 1

(Strongly Disagree) to 5 (Strongly agree). Reliability analysis of usefulness questions shows an acceptable reliability level (alpha > 0.7)

```
relUPost<-data.frame(P_data$Usefulness1, P_data$Usefulness2, P_data$Usefulness3, P_data$Usefulness4, P_d
alpha(relUPost)
```

```
##
## Reliability analysis
## Call: alpha(x = relUPost)
##
##   raw_alpha std.alpha G6(smc) average_r S/N    ase mean   sd median_r
##       0.96      0.96    0.96      0.81  26 0.0044  3.8 0.98     0.84
##
##  lower alpha upper     95% confidence boundaries
## 0.95 0.96 0.97
##
##  Reliability if an item is dropped:
##                     raw_alpha std.alpha G6(smc) average_r S/N alpha se  var.r
## P_data.Usefulness1      0.95      0.95    0.95      0.80  20   0.0058 0.0022
## P_data.Usefulness2      0.96      0.96    0.95      0.82  22   0.0053 0.0019
## P_data.Usefulness3      0.96      0.96    0.95      0.81  22   0.0054 0.0024
## P_data.Usefulness4      0.96      0.96    0.95      0.81  22   0.0054 0.0025
## P_data.Usefulness5      0.96      0.96    0.96      0.84  26   0.0045 0.0011
## P_data.Usefulness6      0.95      0.95    0.95      0.80  20   0.0058 0.0025
##                     med.r
## P_data.Usefulness1  0.79
## P_data.Usefulness2  0.84
## P_data.Usefulness3  0.81
## P_data.Usefulness4  0.82
## P_data.Usefulness5  0.85
## P_data.Usefulness6  0.80
##
##  Item statistics
##                       n raw.r std.r r.cor r.drop mean  sd
## P_data.Usefulness1 174  0.94  0.95  0.94   0.92  3.8 1.0
## P_data.Usefulness2 174  0.91  0.91  0.90   0.87  3.8 1.1
## P_data.Usefulness3 174  0.92  0.92  0.90   0.88  3.9 1.1
## P_data.Usefulness4 174  0.92  0.92  0.91   0.89  3.9 1.0
## P_data.Usefulness5 174  0.88  0.87  0.83   0.82  3.6 1.1
## P_data.Usefulness6 174  0.94  0.94  0.93   0.91  3.8 1.1
##
## Non missing response frequency for each item
##                       1    2    3    4    5 miss
## P_data.Usefulness1 0.04 0.10 0.07 0.56 0.23    0
## P_data.Usefulness2 0.06 0.09 0.10 0.48 0.27    0
## P_data.Usefulness3 0.06 0.08 0.09 0.50 0.28    0
## P_data.Usefulness4 0.03 0.10 0.08 0.52 0.26    0
## P_data.Usefulness5 0.06 0.13 0.16 0.44 0.21    0
## P_data.Usefulness6 0.05 0.09 0.10 0.51 0.25    0
```

```
relUPre<-data.frame(P_data$`Pre-Usefulness1`, P_data$`Pre-Usefulness2`, P_data$`Pre-Usefulness3`, P_data
alpha(relUPre)
```

```
##
```

```
## Reliability analysis
## Call: alpha(x = relUPre)
##
##   raw_alpha std.alpha G6(smc) average_r S/N    ase mean   sd median_r
##       0.95      0.95    0.95      0.76  19 0.0061  3.7 0.83     0.77
##
##  lower alpha upper     95% confidence boundaries
## 0.94 0.95 0.96
##
##  Reliability if an item is dropped:
##                          raw_alpha std.alpha G6(smc) average_r S/N alpha se
## P_data..Pre.Usefulness1.      0.94      0.94    0.93      0.75  15   0.0078
## P_data..Pre.Usefulness2.      0.94      0.94    0.93      0.75  15   0.0080
## P_data..Pre.Usefulness3.      0.95      0.95    0.94      0.79  19   0.0065
## P_data..Pre.Usefulness.4.     0.93      0.94    0.93      0.75  15   0.0080
## P_data..Pre.Usefulness5.      0.95      0.95    0.94      0.79  19   0.0061
## P_data..Pre.Usefulness6.      0.93      0.94    0.93      0.75  15   0.0082
##                           var.r med.r
## P_data..Pre.Usefulness1.  0.0054  0.76
## P_data..Pre.Usefulness2.  0.0055  0.76
## P_data..Pre.Usefulness3.  0.0033  0.81
## P_data..Pre.Usefulness.4. 0.0049  0.74
## P_data..Pre.Usefulness5.  0.0025  0.81
## P_data..Pre.Usefulness6.  0.0058  0.73
##
##  Item statistics
##                             n raw.r std.r r.cor r.drop mean   sd
## P_data..Pre.Usefulness1.  174  0.91  0.92  0.90   0.87  3.8 0.88
## P_data..Pre.Usefulness2.  174  0.92  0.92  0.91   0.88  3.8 0.91
## P_data..Pre.Usefulness3.  174  0.85  0.85  0.80   0.78  3.9 0.96
## P_data..Pre.Usefulness.4. 174  0.92  0.93  0.92   0.89  3.8 0.89
## P_data..Pre.Usefulness5.  174  0.85  0.84  0.79   0.77  3.4 1.04
## P_data..Pre.Usefulness6.  174  0.93  0.93  0.92   0.89  3.8 0.90
##
## Non missing response frequency for each item
##                              1    2    3    4    5 miss
## P_data..Pre.Usefulness1.  0.02 0.11 0.08 0.66 0.13    0
## P_data..Pre.Usefulness2.  0.02 0.10 0.13 0.60 0.15    0
## P_data..Pre.Usefulness3.  0.04 0.05 0.12 0.53 0.26    0
## P_data..Pre.Usefulness.4. 0.02 0.10 0.06 0.67 0.14    0
## P_data..Pre.Usefulness5.  0.05 0.13 0.28 0.41 0.13    0
## P_data..Pre.Usefulness6.  0.02 0.10 0.09 0.64 0.14    0
```

3.2 Data preparation

Since the reliability level was acceptable, we continued with getting a unified score for usefulness. First, we calculated the average of the pre and post questionnaire. Then, we transfer the data into another structure (Id, Group, Session, Score). The new structure can be used to fit a generalized model. After that, we subset the data to the three groups (i.e., chatbot, text support, no support).

```
P_data$PostUsefulness = (P_data$Usefulness1 + P_data$Usefulness2 + P_data$Usefulness3 + P_data$Usefulne

P_data$PreUsefulness = (P_data$`Pre-Usefulness1` + P_data$`Pre-Usefulness2` + P_data$`Pre-Usefulness3` +
```

```r
Utemp = P_data %>% select(ParticipantID,Group, PreUsefulness, PostUsefulness, HighestBarrier)

Utemp= Utemp[order(Utemp$ParticipantID),]
#Utemp

UsfPre <- data.frame("","","","","")
UsfPost <- data.frame("","","","","")


for (i in 1:nrow(Utemp)) { # for loop for transforming the sata
  UsfPre <- data.frame(Utemp$ParticipantID, Utemp$Group, "Pre", Utemp$PreUsefulness, Utemp$HighestBarri

  UsfPost <- data.frame(Utemp$ParticipantID, Utemp$Group, "Post", Utemp$PostUsefulness, Utemp$HighestBa
}

colnames(UsfPre) <- "ParticipantID"
colnames(UsfPre)[2] <- "Group"
colnames(UsfPre)[3] <- "Session"
colnames(UsfPre)[4] <- "Score"
colnames(UsfPre)[5] <- "HighestBarrier"

colnames(UsfPost) <- "ParticipantID"
colnames(UsfPost)[2] <- "Group"
colnames(UsfPost)[3] <- "Session"
colnames(UsfPost)[4] <- "Score"
colnames(UsfPost)[5] <- "HighestBarrier"


Usf <- rbind(UsfPre, UsfPost) # the new data set


Usf$ScoreRverse = 6-Usf$Score # make all values positive and reverse to fit distribution (best is 1, wo




# Subset the data
chatbotUsf= Usf[Usf$Group=="Chatbot Support", ]
noSuppUsf = Usf[Usf$Group=="No Support", ]
txtSuppUsf = Usf[Usf$Group=="Text Support", ]
```

3.3 Assumption checking

Before analysing the data, we checked for distribution normality. This was done visually for the 3 conditions:

```r
stem(chatbotUsf$ScoreRverse)
```

```
##
##   The decimal point is 1 digit(s) to the left of the |
##
##   10 | 00000000000000777
```

```
##   12 | 33
##   14 | 000000
##   16 | 777777
##   18 | 3333333333333
##   20 | 000000000000000000000000000007777777777777
##   22 | 33333
##   24 | 000
##   26 | 777
##   28 | 3
##   30 |
##   32 | 3
##   34 | 0
##   36 | 77
##   38 | 333
##   40 | 0007777
##   42 | 3
##   44 |
##   46 |
##   48 | 3
##   50 | 00
```

```r
stem(txtSuppUsf$ScoreRverse)
```

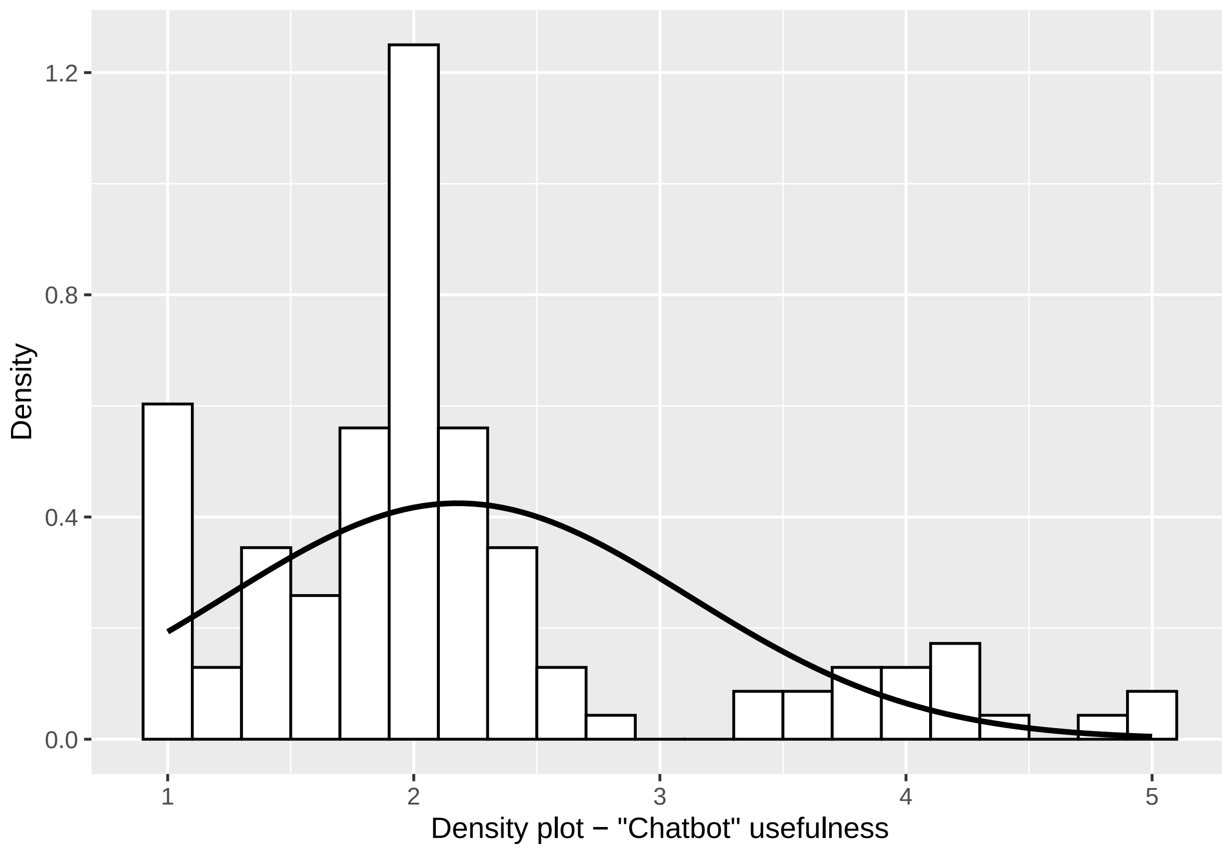
```
##
##   The decimal point is 1 digit(s) to the left of the |
##
##   10 | 0000000000077
##   12 | 3333
##   14 | 00000000
##   16 | 777
##   18 | 3333
##   20 | 000000000000000000000000000000000000000077777777777
##   22 | 33333
##   24 | 0000
##   26 | 77
##   28 | 3333
##   30 | 000007
##   32 | 3
##   34 | 0
##   36 |
##   38 | 33
##   40 | 0000
##   42 | 3
##   44 |
##   46 |
##   48 |
##   50 | 000
```

```r
stem(noSuppUsf$ScoreRverse)
```

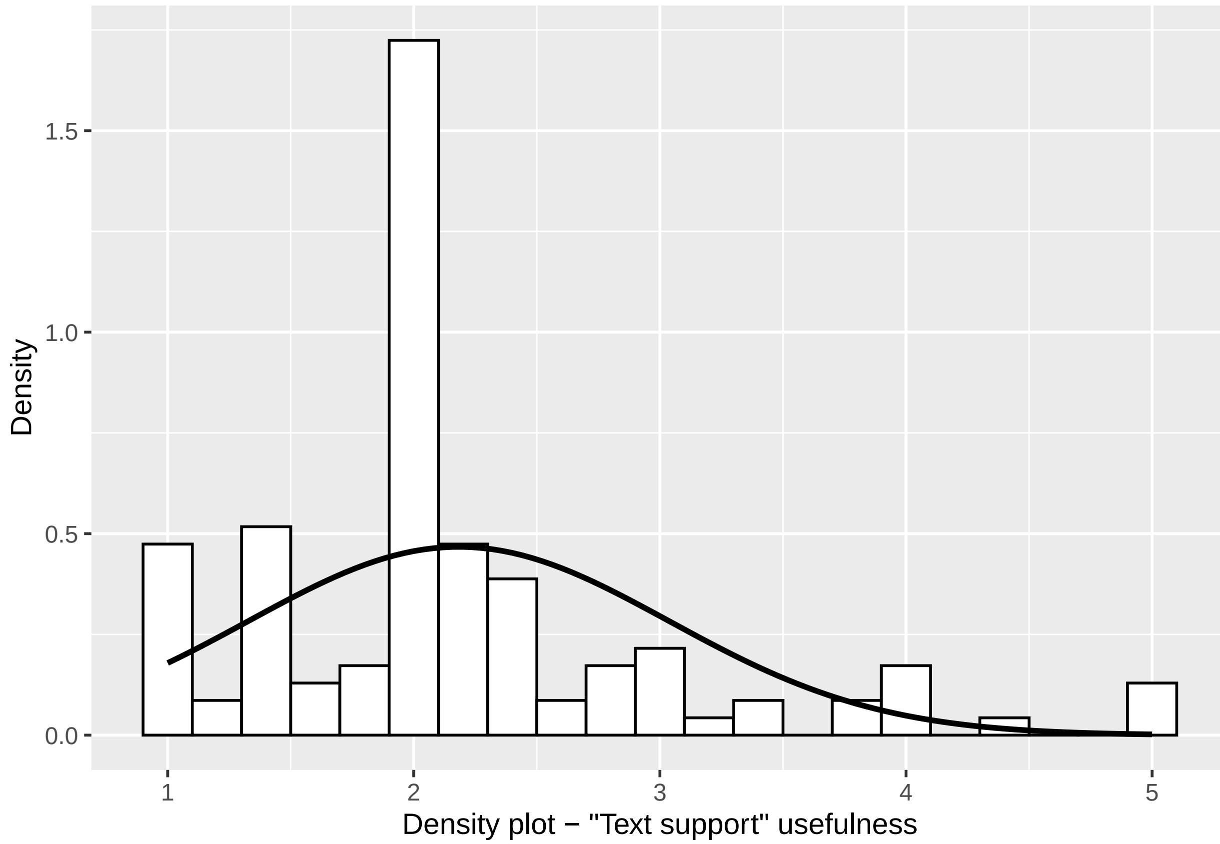
```
##
##   The decimal point is 1 digit(s) to the left of the |
##
```

```
##    10 | 000000007
##    12 | 333
##    14 | 000
##    16 | 7777777777
##    18 | 33333333333333
##    20 | 0000000000000000000000000777777777777777
##    22 | 3333
##    24 | 00000
##    26 | 777
##    28 | 3
##    30 | 00007
##    32 |
##    34 | 000
##    36 |
##    38 | 3
##    40 | 0000007777
##    42 | 3
##    44 |
##    46 | 77
##    48 |
##    50 | 00
```

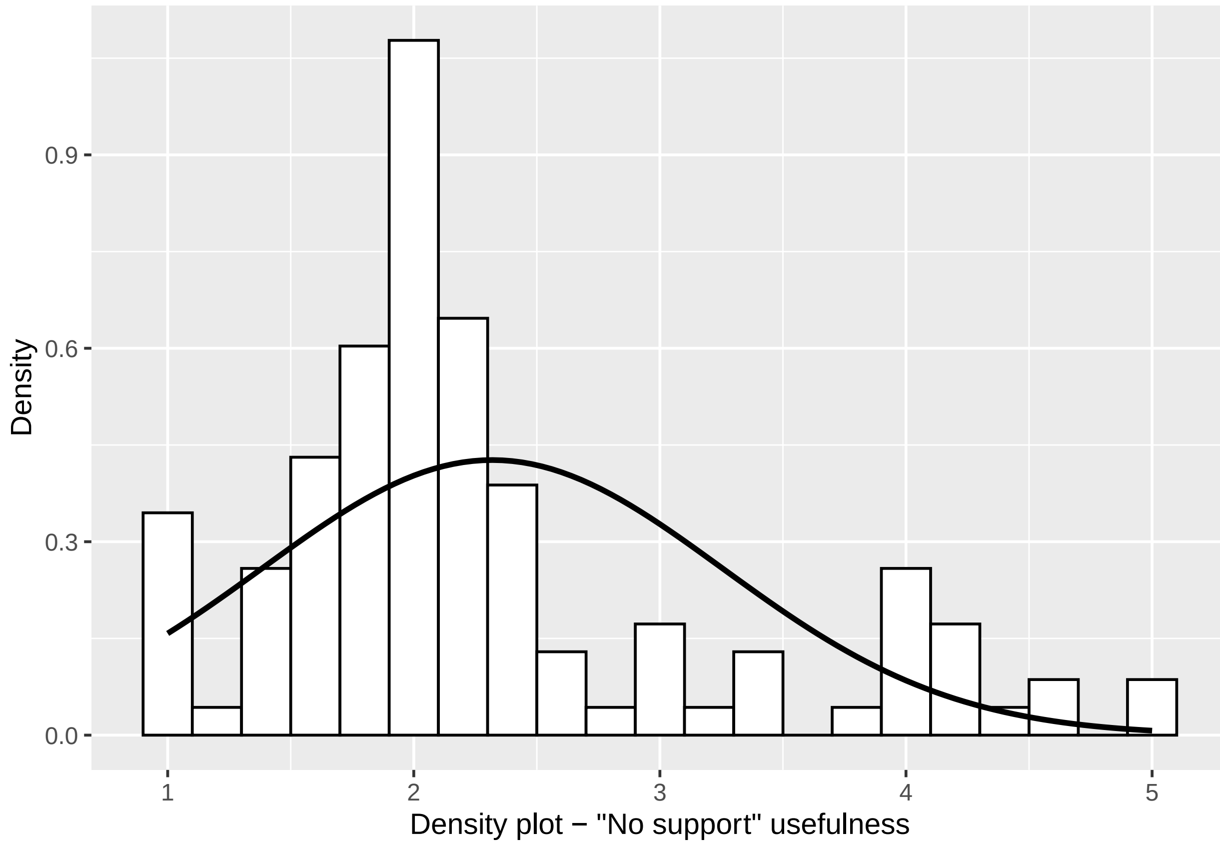
```
ggplot(chatbotUsf,
aes(ScoreRverse)) + geom_histogram(aes(y=..density..),binwidth = 0.2,
colour="black", fill="white") + labs(x='Density plot - "Chatbot" usefulness',
y="Density") + stat_function(fun=dnorm,
args=list(mean=mean(chatbotUsf$ScoreRverse, na.rm=TRUE),
sd=sd(chatbotUsf$ScoreRverse, na.rm=TRUE)), colour="black", size=1)
```




```r
ggplot(txtSuppUsf,
aes(ScoreRverse)) + geom_histogram(aes(y=..density..),binwidth = 0.2,
colour="black", fill="white") + labs(x='Density plot - "Text support" usefulness',
y="Density") + stat_function(fun=dnorm,
args=list(mean=mean(txtSuppUsf$ScoreRverse, na.rm=TRUE),
sd=sd(txtSuppUsf$ScoreRverse, na.rm=TRUE)), colour="black", size=1)
```




```
ggplot(noSuppUsf,
aes(ScoreRverse)) + geom_histogram(aes(y=..density..),binwidth = 0.2,
colour="black", fill="white") + labs(x='Density plot - "No support" usefulness',
y="Density") + stat_function(fun=dnorm,
args=list(mean=mean(noSuppUsf$ScoreRverse, na.rm=TRUE),
sd=sd(noSuppUsf$ScoreRverse, na.rm=TRUE)), colour="black", size=1)
```

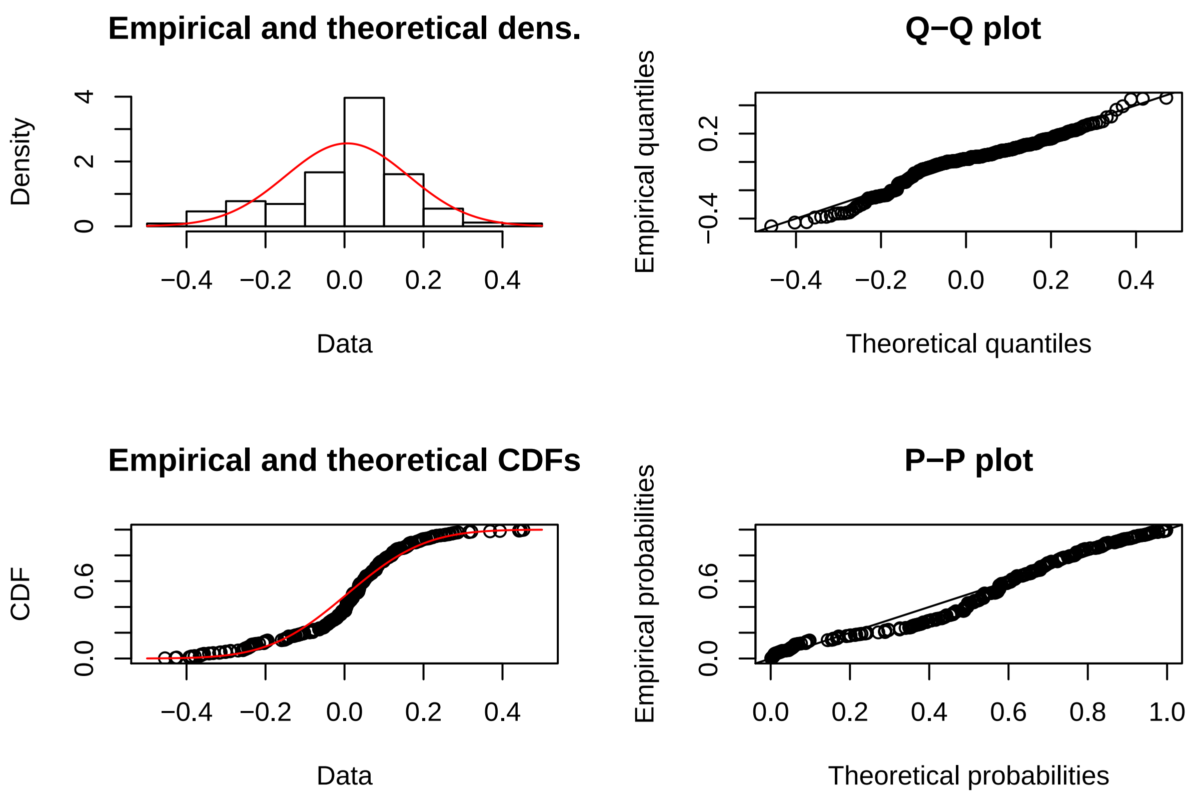


The data in the histograms shows a clear deviation from normal distribution.

3.4 Analysis of data

A generalized multilevel mixed effect model was fitted, wherein as a random effect, we used participant, and as fixed effects, we used the pre and post sessions. The model has a random intercept and a fixed slope, as we are assuming that all participants have the same direction but with various starting points. First, we checked if the residuals fits the distribution in case of using Gamma distribution.

```
mg <- glmer(ScoreRverse ~ Group + Session + Group:Session +
    (1 | ParticipantID), data = Usf, family = Gamma)


fit.g <- fitdist(residuals(mg), "norm")

plot(fit.g)
```



The plots looks reasonable. We continued analysing the data using the same model.

```r
Anova(mg)
```

Table 2: Analysis of Deviance Table (Type II Wald chisquare tests)

	Chisq	Df	Pr(>Chisq)
Group	0.9011	2	0.6373
Session	1.991	1	0.1582
Group:Session	10.38	2	0.00558

```r
summary(mg)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: Gamma  ( inverse )
## Formula: ScoreRverse ~ Group + Session + Group:Session + (1 | ParticipantID)
##    Data: Usf
##
##      AIC      BIC   logLik deviance df.resid
##    486.9    517.7   -235.4    470.9      340
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
```

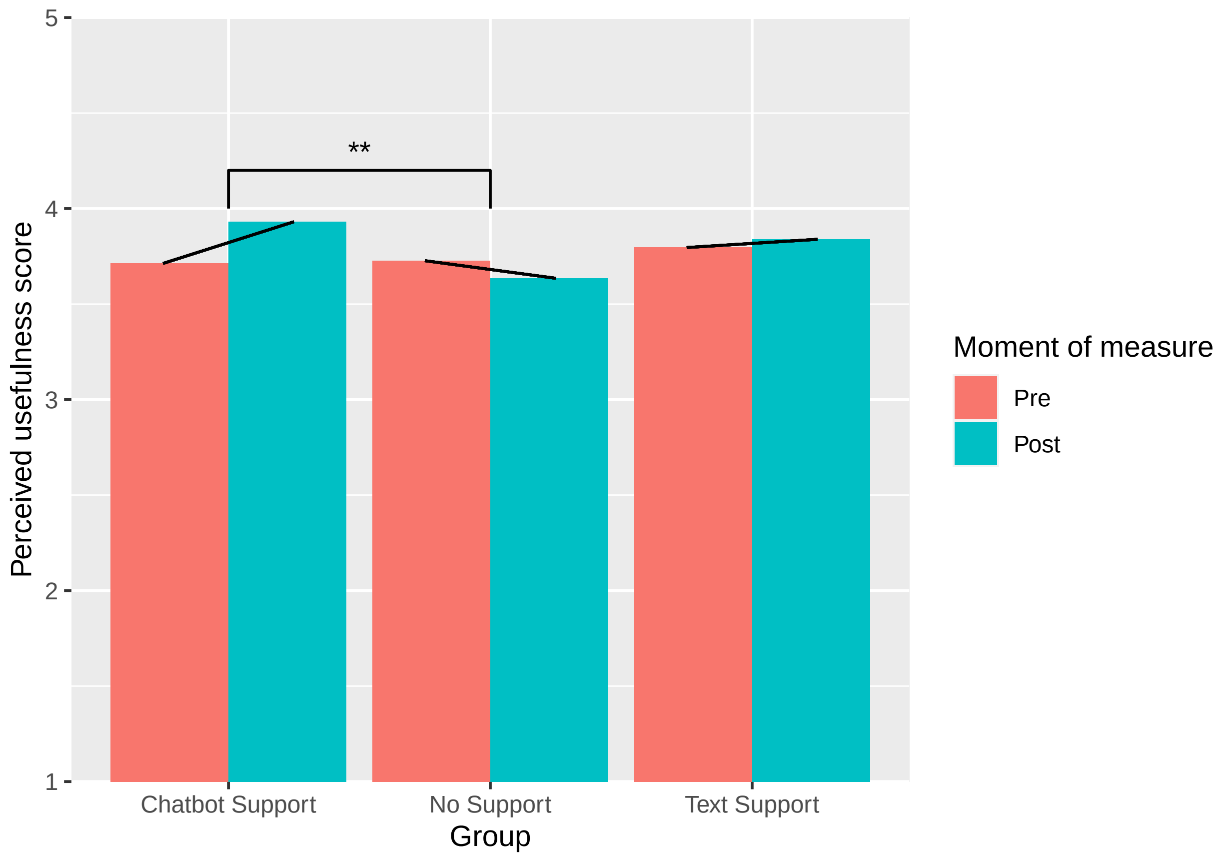
```
## -1.74559 -0.22382  0.09706  0.42520  2.35349
##
## Random effects:
##  Groups        Name        Variance Std.Dev.
##  ParticipantID (Intercept) 0.01931  0.1389
##  Residual                  0.04952  0.2225
## Number of obs: 348, groups:  ParticipantID, 174
##
## Fixed effects:
##                                  Estimate Std. Error t value Pr(>|z|)
## (Intercept)                      0.561355   0.029972  18.730  < 2e-16 ***
## GroupNo Support                 -0.012614   0.042042  -0.300  0.76416
## GroupText Support               -0.006847   0.041603  -0.165  0.86928
## SessionPost                      0.039755   0.012382   3.211  0.00132 **
## GroupNo Support:SessionPost     -0.054711   0.017010  -3.216  0.00130 **
## GroupText Support:SessionPost -0.031695   0.017599  -1.801  0.07171 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) GrpNSp GrpTxS SssnPs GNS:SP
## GroupNSpprt -0.685
## GrpTxtSpprt -0.690  0.490
## SessionPost -0.188  0.133  0.135
## GrpNSppr:SP  0.138 -0.196 -0.098 -0.728
## GrpTSppr:SP  0.132 -0.094 -0.199 -0.704  0.512
## convergence code: 0
## Model failed to converge with max|grad| = 0.00251212 (tol = 0.001, component 1)
```

the interaction between the groups and the sessions shows a significance $p < 0.01$. Therefore, there is a difference between the groups. Also, the usefulness shows a significant p-value between the the chatbot and no support ($p<0.01$)

The following bar chart show the difference between the pre and post questionnaire means for the 3 conditions.

```
# Subset the data

chatbot= P_data[P_data$Group=="Chatbot Support", ]
noSupp = P_data[P_data$Group=="No Support", ]
txtSupp = P_data[P_data$Group=="Text Support", ]


path1 = data.frame(x=c(1,1,2,2),y=c(4,4.2,4.2,4))
#path2 = data.frame(x2=c(1,1,3,3),y2=c(4.4,4.6,4.6,4.4))
y1cU=mean(chatbot$PreUsefulness)
y2cU=mean(chatbot$PostUsefulness)
y1nU=mean(noSupp$PreUsefulness)
y2nU=mean(noSupp$PostUsefulness)
y1tU=mean(txtSupp$PreUsefulness)
y2tU=mean(txtSupp$PostUsefulness)




by_Usf <- Usf %>% group_by(Group, Session) %>% summarise(Score = mean(Score))
```

```r
ggplot(by_Usf) +
    geom_bar(aes(x = Group, y=Score, fill = Session), position = position_dodge(preserve = 'single'), s
```



4 Self-Efficacy analysis

4.1 Reliability check

The participant in all three conditions were asked to fill a self-efficacy questionnaire of before and after doing the exercise. The questionnaire includes 5 questions which they were asked to rate from 0 (highly certain cannot do) to 10 (highly certain can do). Reliability analysis of self-efficacy questions shows an acceptable reliability level (alpha > 0.7)

```r
relUPost<-data.frame(P_data$`Self-Efficacy1`, P_data$`Self-Efficacy2`, P_data$`Self-Efficacy3`, P_data$
alpha(relUPost)
```

```
##
## Reliability analysis
## Call: alpha(x = relUPost)
##
##   raw_alpha std.alpha G6(smc) average_r S/N   ase mean  sd median_r
##       0.79      0.79    0.77      0.43 3.7 0.024  5.6 2.1     0.41
##
```

```
##  lower alpha upper     95% confidence boundaries
## 0.74 0.79 0.84
##
##  Reliability if an item is dropped:
##                           raw_alpha std.alpha G6(smc) average_r S/N alpha se
## P_data..Self.Efficacy1.      0.79      0.79    0.75      0.48 3.7    0.026
## P_data..Self.Efficacy2.      0.73      0.73    0.70      0.41 2.8    0.032
## P_data..Self.Efficacy3.      0.72      0.72    0.70      0.39 2.6    0.034
## P_data..Self.Efficacy4.      0.74      0.75    0.71      0.42 2.9    0.031
## P_data..Self.Efficacy5.      0.76      0.75    0.72      0.43 3.0    0.026
##                           var.r med.r
## P_data..Self.Efficacy1. 0.011  0.50
## P_data..Self.Efficacy2. 0.013  0.41
## P_data..Self.Efficacy3. 0.017  0.36
## P_data..Self.Efficacy4. 0.011  0.41
## P_data..Self.Efficacy5. 0.024  0.45
##
##  Item statistics
##                           n raw.r std.r r.cor r.drop mean  sd
## P_data..Self.Efficacy1. 174  0.58  0.65  0.51   0.44  8.7 1.9
## P_data..Self.Efficacy2. 174  0.79  0.77  0.70   0.63  4.2 3.0
## P_data..Self.Efficacy3. 174  0.82  0.79  0.73   0.66  4.3 3.2
## P_data..Self.Efficacy4. 174  0.77  0.74  0.67   0.60  4.6 3.0
## P_data..Self.Efficacy5. 174  0.72  0.73  0.63   0.53  6.2 2.9
```

```r
relUPre<-data.frame(P_data$`Pre-Self-Efficacy1`, P_data$`Pre-Self-Efficacy-2`, P_data$`Pre-Self-Efficac
alpha(relUPre)
```

```
##
## Reliability analysis
## Call: alpha(x = relUPre)
##
##   raw_alpha std.alpha G6(smc) average_r S/N   ase mean sd median_r
##       0.75      0.75    0.72      0.37 2.9 0.029  5.4  2      0.4
##
##  lower alpha upper     95% confidence boundaries
## 0.69 0.75 0.81
##
##  Reliability if an item is dropped:
##                                raw_alpha std.alpha G6(smc) average_r S/N alpha se
## P_data..Pre.Self.Efficacy1.       0.75      0.75    0.70      0.43 3.0    0.031
## P_data..Pre.Self.Efficacy.2.      0.69      0.69    0.65      0.36 2.3    0.037
## P_data..Pre.Self.Efficacy.3.      0.66      0.67    0.63      0.33 2.0    0.040
## P_data..Pre.Self.Efficacy.4.      0.70      0.70    0.67      0.37 2.4    0.035
## P_data..Pre.Self.Efficacy.5.      0.70      0.69    0.65      0.36 2.2    0.034
##                                 var.r med.r
## P_data..Pre.Self.Efficacy1.  0.0054  0.44
## P_data..Pre.Self.Efficacy.2. 0.0132  0.40
## P_data..Pre.Self.Efficacy.3. 0.0119  0.35
## P_data..Pre.Self.Efficacy.4. 0.0134  0.38
## P_data..Pre.Self.Efficacy.5. 0.0212  0.36
##
##  Item statistics
##                                n raw.r std.r r.cor r.drop mean  sd
```

```
## P_data..Pre.Self.Efficacy1.  174  0.53  0.60  0.45   0.37  8.6 1.9
## P_data..Pre.Self.Efficacy.2. 174  0.75  0.72  0.63   0.55  4.0 3.0
## P_data..Pre.Self.Efficacy.3. 174  0.79  0.77  0.70   0.62  4.2 3.0
## P_data..Pre.Self.Efficacy.4. 174  0.73  0.70  0.59   0.52  4.5 3.1
## P_data..Pre.Self.Efficacy.5. 174  0.71  0.73  0.64   0.53  5.8 2.8
```

4.2 Data preparation

Since the reliability level was acceptable, we continued with getting a unified score for self-efficacy First, we calculated the average of the pre and post questionnaire. Then, we transfer the data into another structure (Id, Group, Session, Score). The new structure can be used to fit a generalized model. After that, we subset the data to the three groups (i.e., chatbot, text support, no support).

```
P_data$PostSE = (P_data$`Self-Efficacy1` + P_data$`Self-Efficacy2` + P_data$`Self-Efficacy3` + P_data$`S

P_data$PreSE = (P_data$`Pre-Self-Efficacy1` + P_data$`Pre-Self-Efficacy-2` + P_data$`Pre-Self-Efficacy-3


SEtemp = P_data %>% select(ParticipantID,Group, PreSE, PostSE)

SEtemp= SEtemp[order(SEtemp$ParticipantID),]
#SEtemp

SeEfPre <- data.frame("","","","")
SeEfPost <- data.frame("","","","")


for (i in 1:nrow(SEtemp)) {
  SeEfPre <- data.frame(SEtemp$ParticipantID, SEtemp$Group, "Pre", SEtemp$PreSE)

  SeEfPost <- data.frame(SEtemp$ParticipantID, SEtemp$Group, "Post", SEtemp$PostSE)
}

colnames(SeEfPre) <- "ParticipantID"
colnames(SeEfPre)[2] <- "Group"
colnames(SeEfPre)[3] <- "Session"
colnames(SeEfPre)[4] <- "Score"

colnames(SeEfPost) <- "ParticipantID"
colnames(SeEfPost)[2] <- "Group"
colnames(SeEfPost)[3] <- "Session"
colnames(SeEfPost)[4] <- "Score"


SeEf <- rbind(SeEfPre, SeEfPost)

SeEf$ScoreRverse = 11 - SeEf$Score # make all values positive and reverse to fit distribution (best is



# Subset the data
chatbotSeEf= SeEf[SeEf$Group=="Chatbot Support", ]
noSuppSeEf = SeEf[SeEf$Group=="No Support", ]
```

```r
txtSuppSeEf = SeEf[SeEf$Group=="Text Support", ]
```

4.3 Assumption checking

Before analysing the data, we checked for distribution normality. This was done visually for the 3 conditions:

```r
stem(chatbotSeEf$ScoreRverse)
```

```
##
##   The decimal point is at the |
##
##   1 | 0000
##   1 | 88
##   2 | 0004
##   2 | 6688888
##   3 | 000222224444
##   3 | 6668888
##   4 | 00022444444
##   4 | 6668888888
##   5 | 0000000444
##   5 | 666666688
##   6 | 0000222224
##   6 | 6666888
##   7 | 00000224
##   7 | 6688
##   8 | 00022
##   8 |
##   9 | 22224
##   9 | 6
```

```r
stem(txtSuppSeEf$ScoreRverse)
```

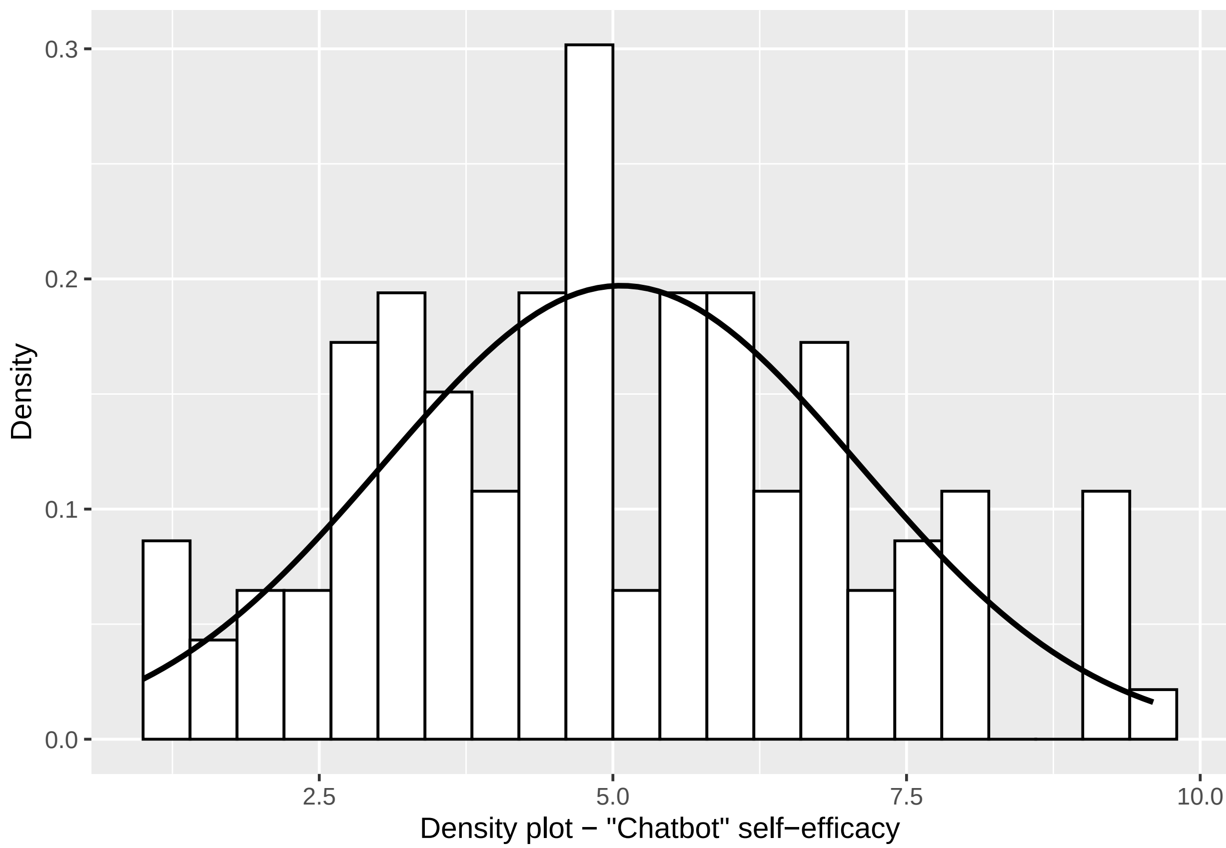
```
##
##   The decimal point is at the |
##
##    1 | 2
##    1 | 666
##    2 | 4
##    2 | 688
##    3 | 00222444
##    3 | 6888
##    4 | 0022224444
##    4 | 68888
##    5 | 000022222444
##    5 | 66666688
##    6 | 0002222222444444444
##    6 | 6666666666888
##    7 | 00222444
##    7 | 6668
##    8 | 00002444
##    8 | 6668
```

```
##    9 | 002
##    9 | 8
##   10 |
##   10 | 8
```

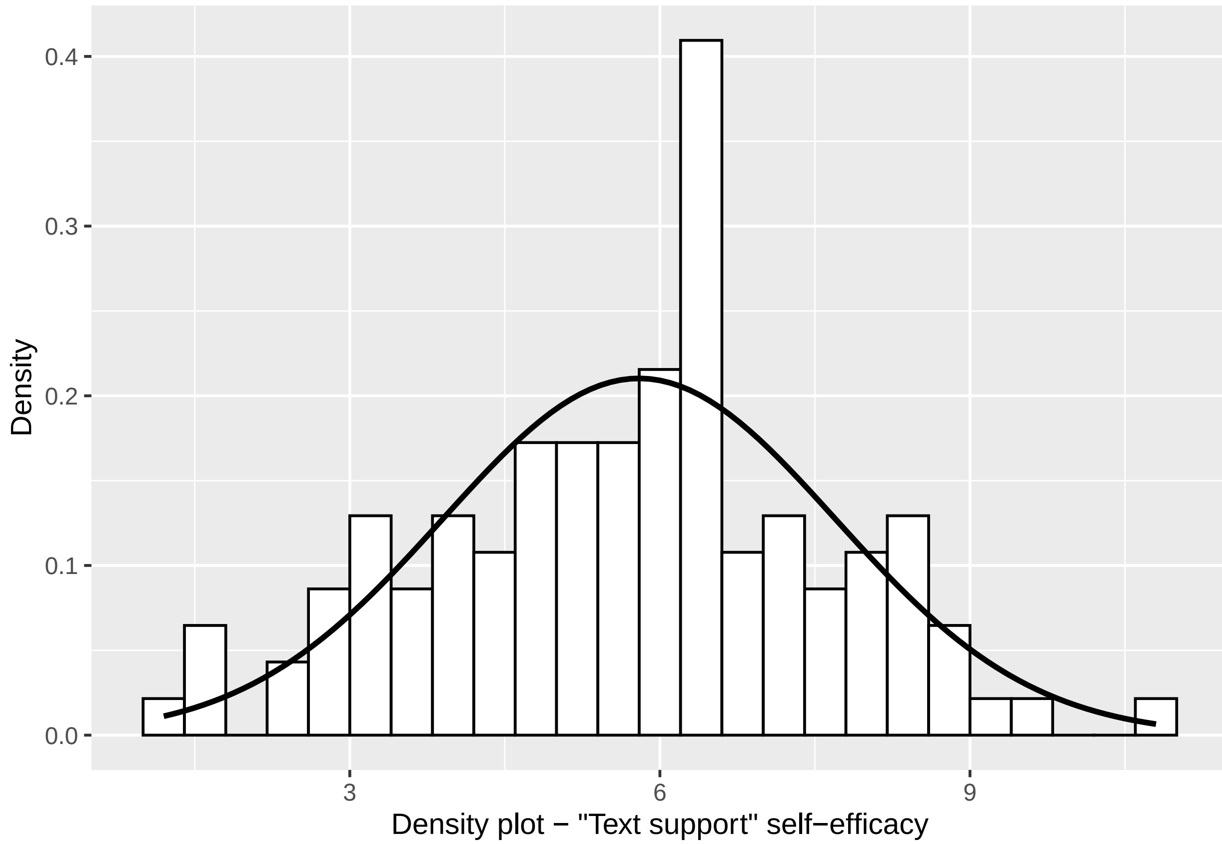
```r
stem(noSuppSeEf$ScoreRverse)
```

```
##
##   The decimal point is at the |
##
##    1 | 004
##    1 | 688
##    2 | 244
##    2 | 68
##    3 | 0222444
##    3 | 666668888
##    4 | 00044444
##    4 | 666888888
##    5 | 00222222224444
##    5 | 666888
##    6 | 00002222222244444
##    6 | 6888888
##    7 | 2222444444
##    7 | 6666688
##    8 | 0
##    8 | 68
##    9 | 0
##    9 | 68
##   10 | 02
##   10 | 688
```

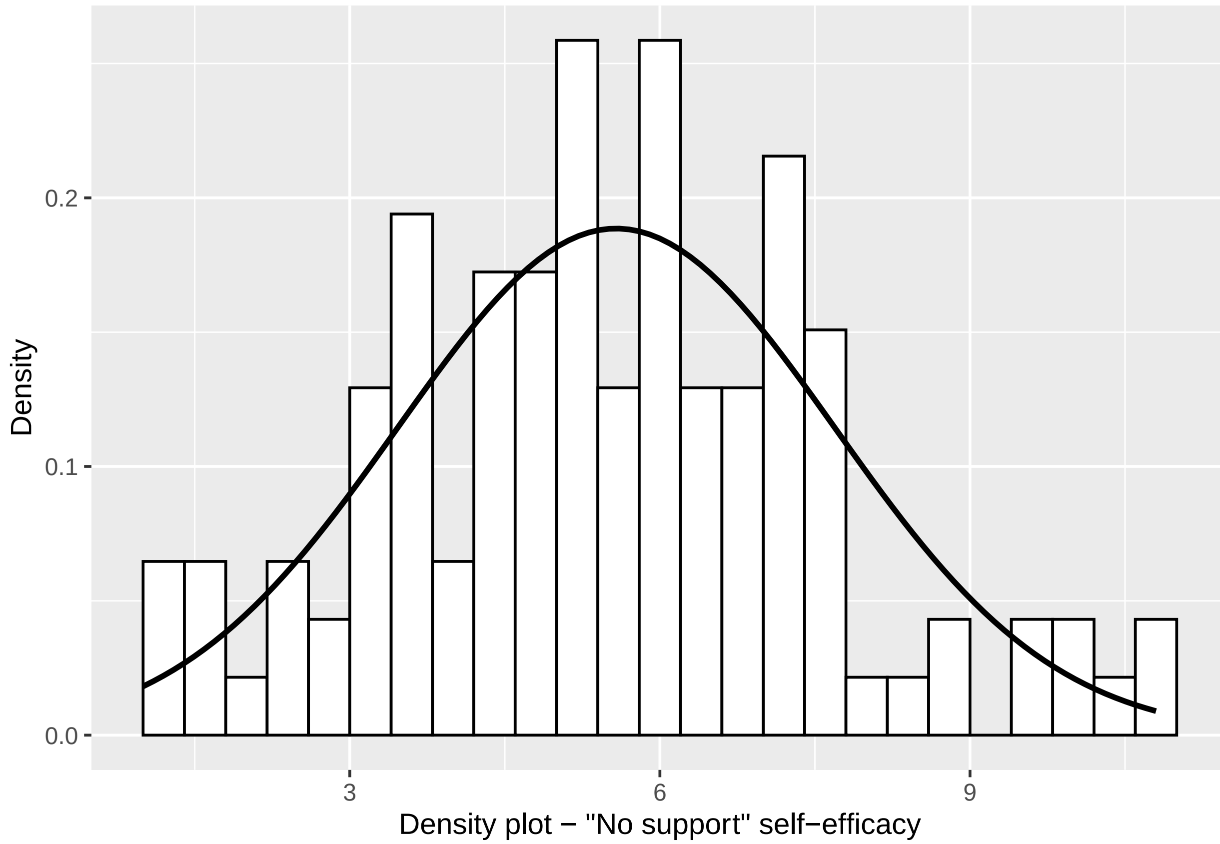
```r
ggplot(chatbotSeEf,
aes(ScoreRverse)) + geom_histogram(aes(y=..density..),binwidth = 0.4,
colour="black", fill="white") + labs(x='Density plot - "Chatbot" self-efficacy',
y="Density") + stat_function(fun=dnorm,
args=list(mean=mean(chatbotSeEf$ScoreRverse, na.rm=TRUE),
sd=sd(chatbotSeEf$ScoreRverse, na.rm=TRUE)), colour="black", size=1)
```




```
ggplot(txtSuppSeEf,
aes(ScoreRverse)) + geom_histogram(aes(y=..density..),binwidth = 0.4,
colour="black", fill="white") + labs(x='Density plot - "Text support" self-efficacy',
y="Density") + stat_function(fun=dnorm,
args=list(mean=mean(txtSuppSeEf$ScoreRverse, na.rm=TRUE),
sd=sd(txtSuppSeEf$ScoreRverse, na.rm=TRUE)), colour="black", size=1)
```




```
ggplot(noSuppSeEf,
aes(ScoreRverse)) + geom_histogram(aes(y=..density..),binwidth = 0.4,
colour="black", fill="white") + labs(x='Density plot - "No support" self-efficacy',
y="Density") + stat_function(fun=dnorm,
args=list(mean=mean(noSuppSeEf$ScoreRverse, na.rm=TRUE),
sd=sd(noSuppSeEf$ScoreRverse, na.rm=TRUE)), colour="black", size=1)
```

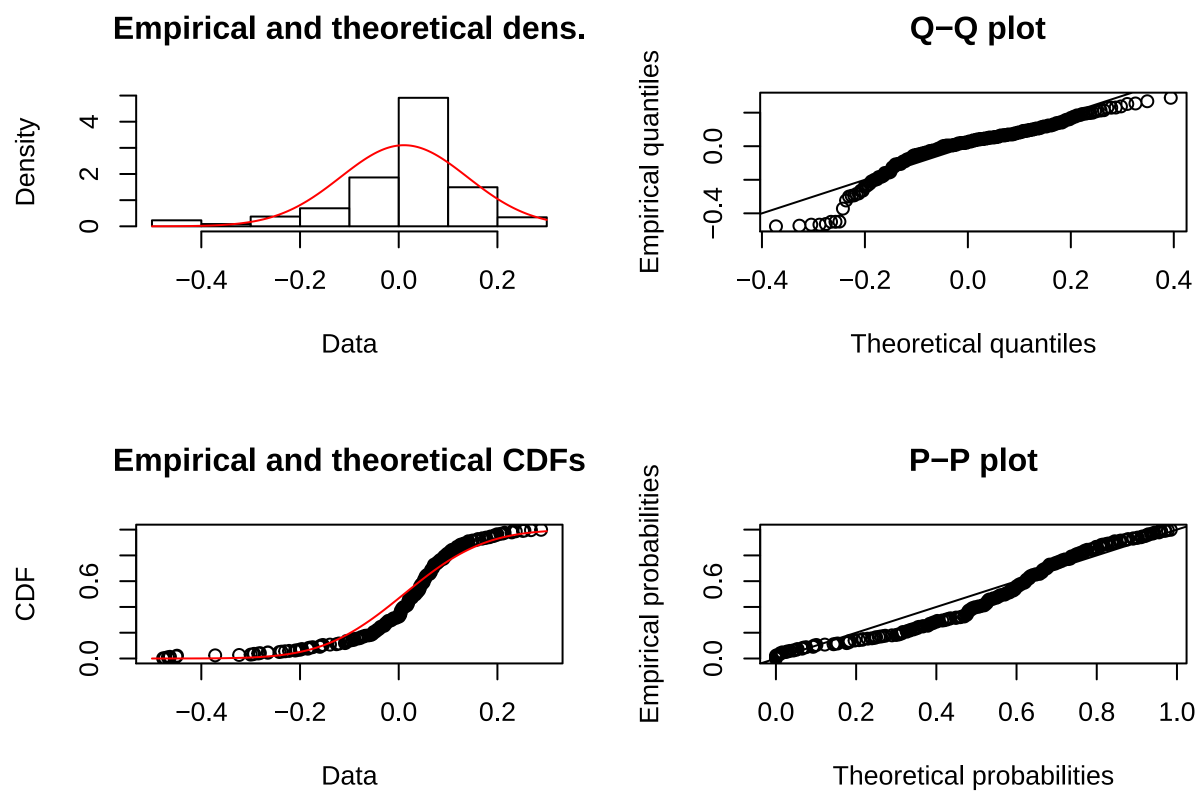


The data in the histograms shows a clear deviation from normal distribution.

4.4 Analysis of data

A generalized multilevel mixed effect model was fitted, wherein as a random effect, we used participant, and as fixed effects, we used the pre and post sessions. The model has a random intercept and a fixed slope, as we are assuming that all participants have the same direction but with various starting points. First, we checked if the residuals fits the distribution in case of using Gamma distribution.

```
SE1 <- glmer(ScoreRverse ~ Group + Session + Group:Session +
    (1 | ParticipantID), data = SeEf, family = Gamma)
fit.SE1 <- fitdist(residuals(SE1), "norm")
plot(fit.SE1)
```



The plots looks reasonable. We continued analysing the data using the same model.

```r
summary(SE1)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: Gamma  ( inverse )
## Formula: ScoreRverse ~ Group + Session + Group:Session + (1 | ParticipantID)
##    Data: SeEf
##
##      AIC      BIC   logLik deviance df.resid
##   1157.0   1187.8   -570.5   1141.0      340
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.8790 -0.1517  0.1597  0.3827  1.4714
##
## Random effects:
##  Groups        Name        Variance Std.Dev.
##  ParticipantID (Intercept) 0.01046  0.1023
##  Residual                  0.04637  0.2153
## Number of obs: 348, groups:  ParticipantID, 174
##
## Fixed effects:
##                                Estimate Std. Error t value Pr(>|z|)
## (Intercept)                    0.311775   0.026071  11.958  < 2e-16 ***
```

```
## GroupNo Support               -0.021373   0.035912  -0.595   0.55174
## GroupText Support             -0.056134   0.037009  -1.517   0.12933
## SessionPost                    0.012156   0.004414   2.754   0.00589 **
## GroupNo Support:SessionPost   -0.009530   0.005971  -1.596   0.11045
## GroupText Support:SessionPost -0.007412   0.005912  -1.254   0.20995
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) GrpNSp GrpTxS SssnPs GNS:SP
## GroupNSpprt -0.653
## GrpTxtSpprt -0.660  0.457
## SessionPost -0.079  0.057  0.055
## GrpNSppr:SP  0.058 -0.079 -0.041 -0.739
## GrpTSppr:SP  0.059 -0.042 -0.075 -0.747  0.552
```

```
Anova(SE1)
```

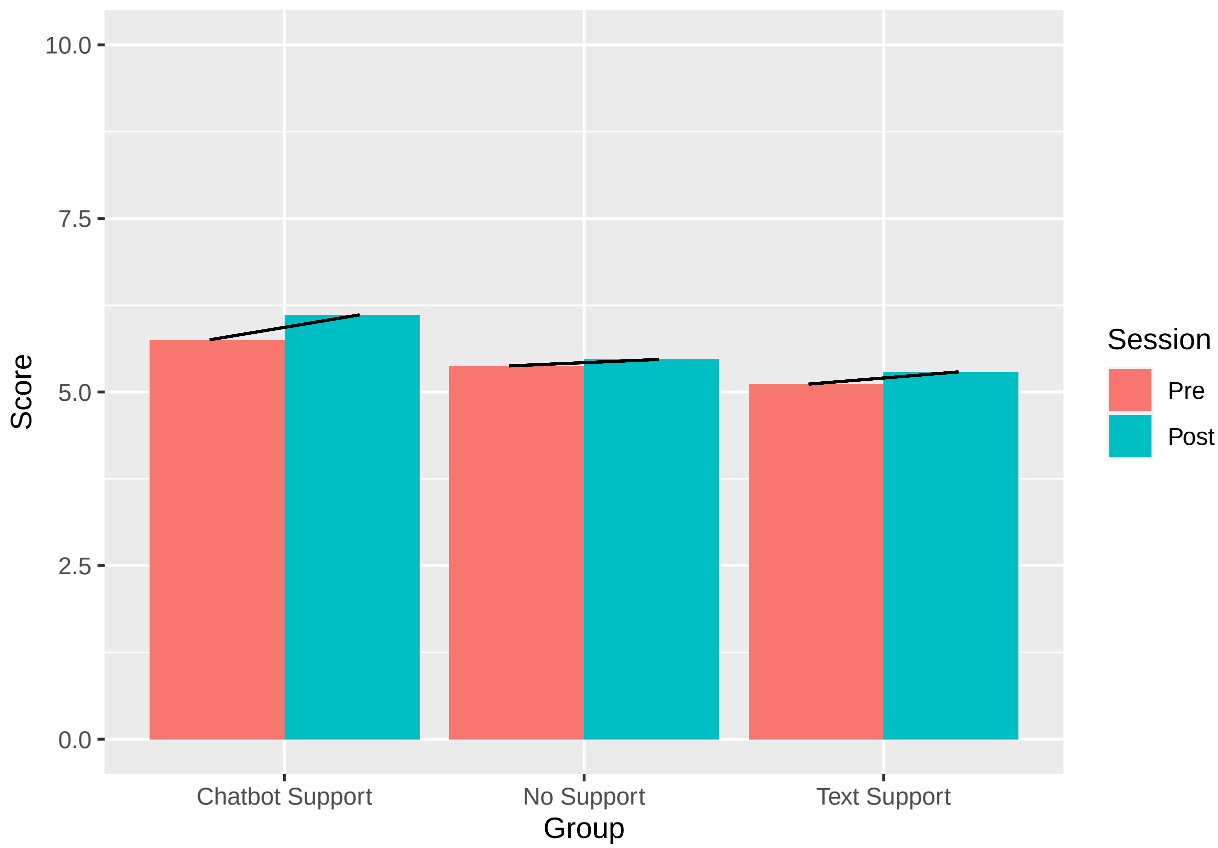
Table 3: Analysis of Deviance Table (Type II Wald chisquare tests)

	Chisq	Df	Pr(>Chisq)
Group	2.609	2	0.2714
Session	6.718	1	0.009544
Group:Session	2.747	2	0.2532

The self-efficacy between the groupd does not show a significant difference ($p>0.05$). However, there is a significant increase between the pre and post measurements ($p<0.05$).

The following bar chart shows the difference between the pre and post questionnaire means for the 3 conditions.

```
# subset the data

chatbot= P_data[P_data$Group=="Chatbot Support", ]
noSupp = P_data[P_data$Group=="No Support", ]
txtSupp = P_data[P_data$Group=="Text Support", ]


pathSE = data.frame(x3=c(1,1,2,2),y3=c(6.3,6.4,6.4,6.3))

y1cU=mean(chatbot$PreSE)
y2cU=mean(chatbot$PostSE)
y1nU=mean(noSupp$PreSE)
y2nU=mean(noSupp$PostSE)
y1tU=mean(txtSupp$PreSE)
y2tU=mean(txtSupp$PostSE)


by_SeEf <- SeEf %>% group_by(Group, Session) %>% summarise(Score = mean(Score))

ggplot(by_SeEf) +
    geom_bar(aes(x = Group, y=Score, fill = Session), position = position_dodge(preserve = 'single'), s
```



4.5 Correlation between perceived usefulness and self-efficacy

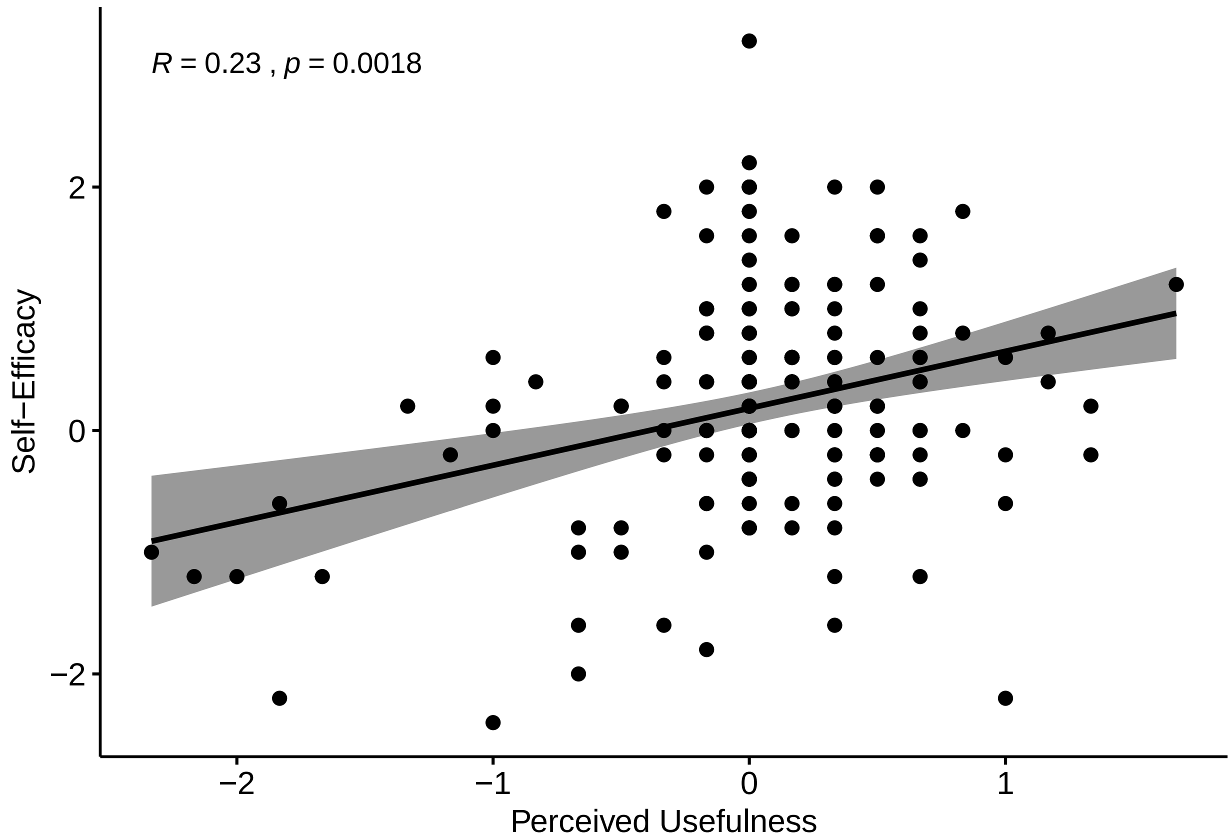
To check if the perceived usefulness and self-efficacy have correlation, Spearman correlation was applied because the data is not normally distributed. We checked the correlation between the delta of both measures (i.e., the difference between post and pre score). The results shows a positive correlation between the measures with $p<0.05$.

```
P_data$UDelta = P_data$PostUsefulness - P_data$PreUsefulness # Get usefulness delta
P_data$SEDelta = P_data$PostSE - P_data$PreSE # Get Self-efficacy delta

cor.test(P_data$UDelta, P_data$SEDelta, method=c("spearman"))
```

Table 4: Spearman's rank correlation rho: `P_data$UDelta` and `P_data$SEDelta`

Test statistic	P value	Alternative hypothesis	rho
672055	0.001839 * *	two.sided	0.2345

```
ggscatter(P_data, x = "UDelta", y = "SEDelta",
          add = "reg.line", conf.int = TRUE,
          cor.coef = TRUE, cor.method = "spearman",
          xlab = "Perceived Usefulness", ylab = "Self-Efficacy")
```



5 Explorative measures

5.1 Enjoyment

We have two measures for enjoyment: the first is a pre and post question were they ask them their prefrence to either do picture labeling task (0) or do the exercise (20). The other measure is a post enjoyment questionnaire which consists of 7 items, and the participants were asked to rank each from 1 (strongly disagree) to 5 (strongly agree). This questionnaire was only presented to chatbot group and text support group.

5.1.1 Enjoyment question

Prepare the data by getting the delta between post and pre.

```
P_data$enjQuestion = P_data$Compare - P_data$`Pre-Compare`

chatbot= P_data[P_data$Group=="Chatbot Support", ]
noSupp = P_data[P_data$Group=="No Support", ]
txtSupp = P_data[P_data$Group=="Text Support", ]
```

checking for normality

```r
stem(chatbot$enjQuestion)
```

```
##
##   The decimal point is 1 digit(s) to the right of the |
##
##   -1 | 300
##   -0 | 8
##   -0 | 2211
##    0 | 00000000000000000000111222233333344
##    0 | 56666666679
##    1 | 00
##    1 |
##    2 | 0
```

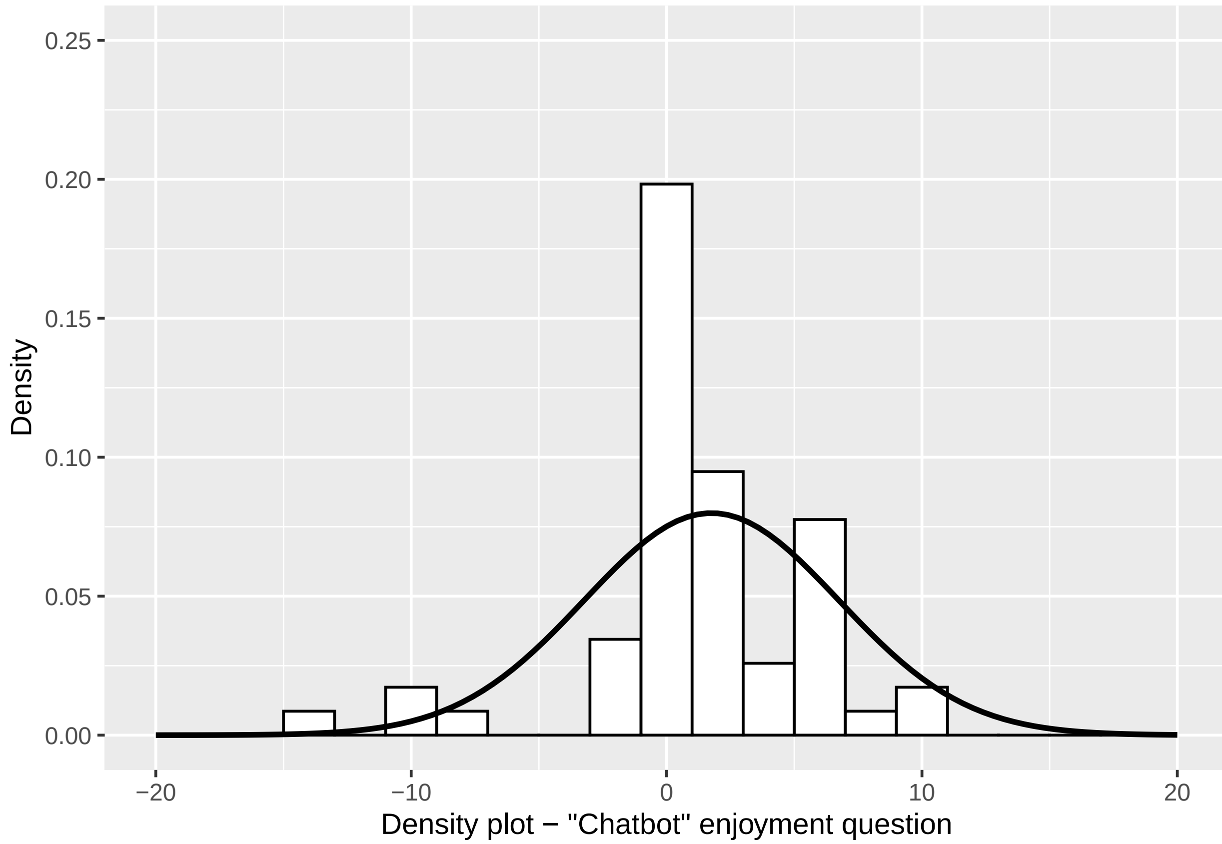
```r
stem(txtSupp$enjQuestion)
```

```
##
##   The decimal point is at the |
##
##   -6 | 00
##   -4 | 0
##   -2 | 0000
##   -0 | 000
##    0 | 00000000000000000000
##    2 | 0000000000
##    4 | 0000000
##    6 | 000000
##    8 | 00
##   10 | 0
##   12 | 0
##   14 | 0
```

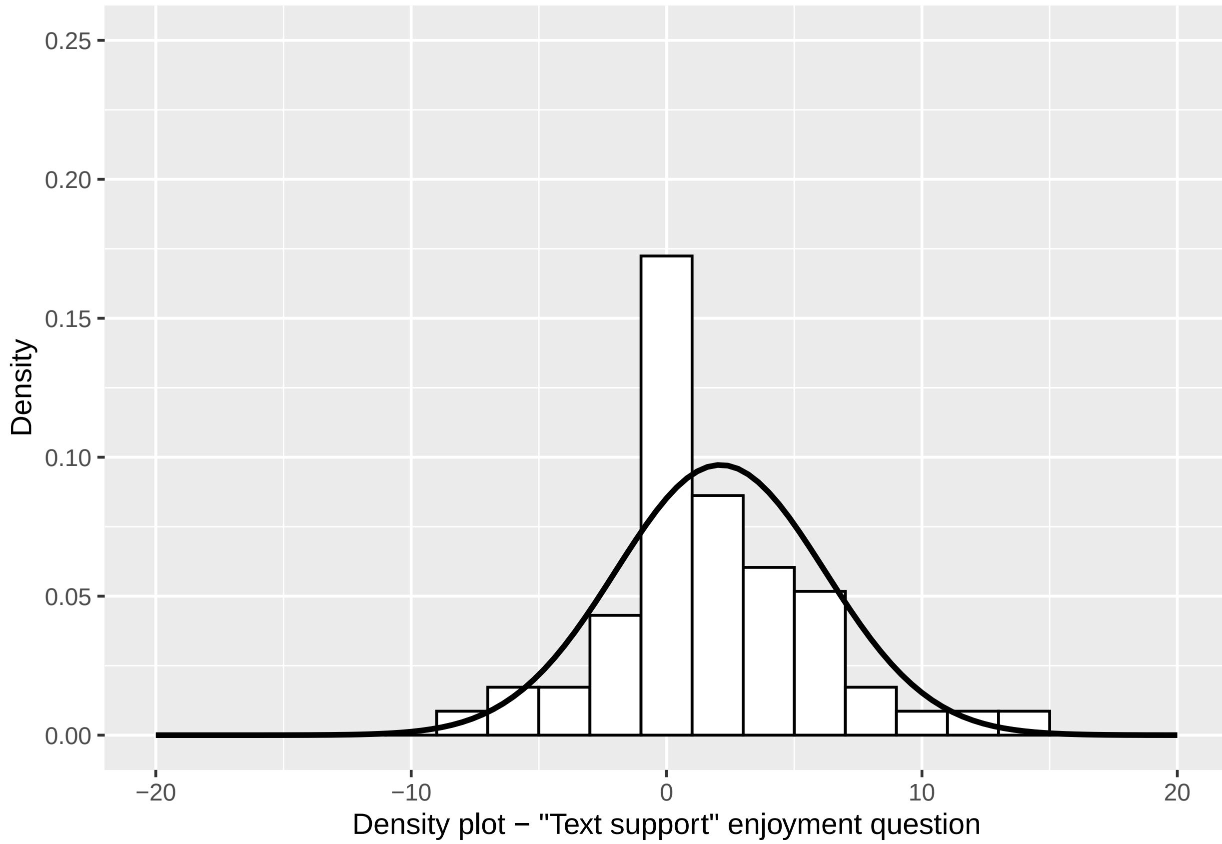
```r
stem(noSupp$enjQuestion)
```

```
##
##   The decimal point is 1 digit(s) to the right of the |
##
##   -1 | 6
##   -1 |
##   -0 | 65
##   -0 | 44433322111
##    0 | 0000000000000000000111111222222233
##    0 | 566789
##    1 | 22
##    1 | 5
```

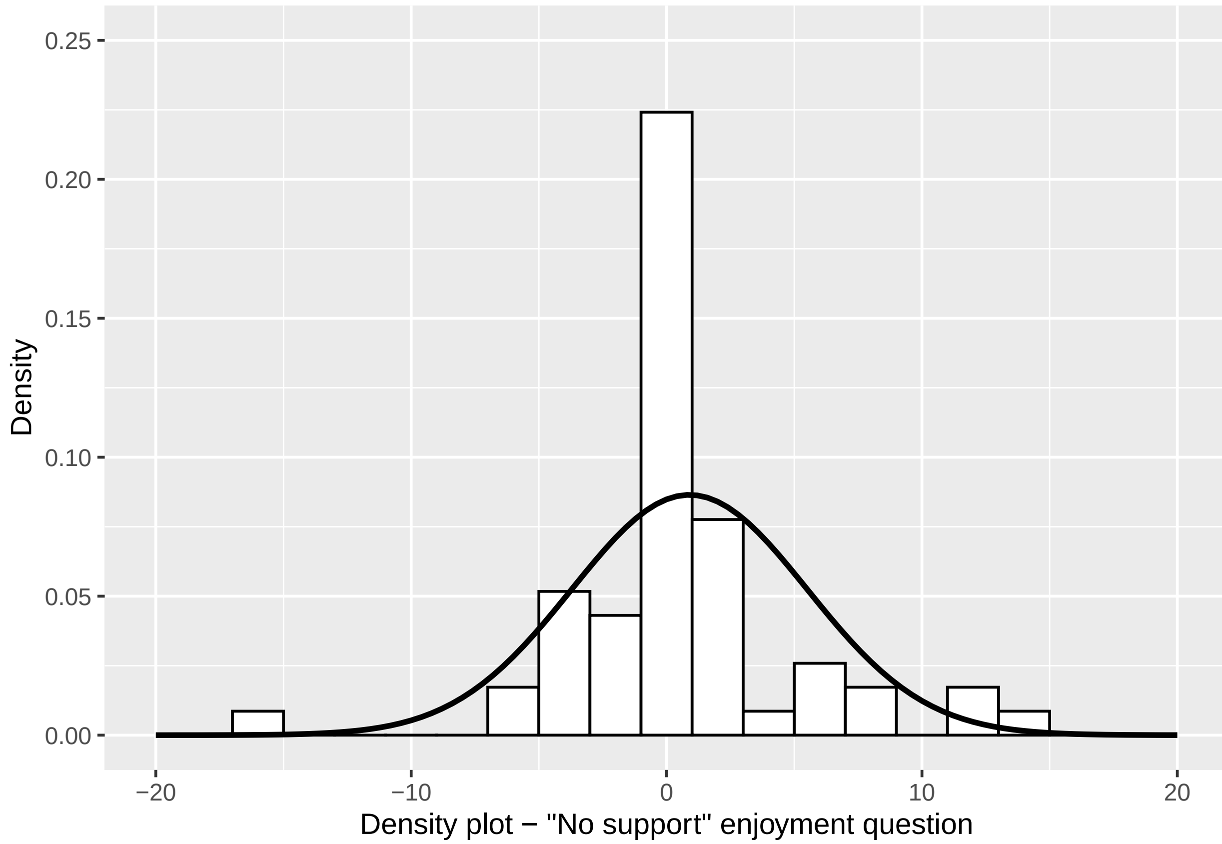
```r
ggplot(chatbot,
aes(enjQuestion)) + geom_histogram(aes(y=..density..),binwidth = 2,
colour="black", fill="white") + labs(x='Density plot - "Chatbot" enjoyment question',
y="Density") + stat_function(fun=dnorm,
args=list(mean=mean(chatbot$enjQuestion, na.rm=TRUE),
sd=sd(chatbot$enjQuestion, na.rm=TRUE)), colour="black", size=1)+ xlim(-20 , 20)+ ylim(0,0.25)
```




```
ggplot(txtSupp,
aes(enjQuestion)) + geom_histogram(aes(y=..density..),binwidth = 2,
colour="black", fill="white") + labs(x='Density plot - "Text support" enjoyment question',
y="Density") + stat_function(fun=dnorm,
args=list(mean=mean(txtSupp$enjQuestion, na.rm=TRUE),
sd=sd(txtSupp$enjQuestion, na.rm=TRUE)), colour="black", size=1)+ xlim(-20 , 20)+ ylim(0,0.25)
```




```
ggplot(noSupp,
aes(enjQuestion)) + geom_histogram(aes(y=..density..),binwidth = 2,
colour="black", fill="white") + labs(x='Density plot - "No support" enjoyment question',
y="Density") + stat_function(fun=dnorm,
args=list(mean=mean(noSupp$enjQuestion, na.rm=TRUE),
sd=sd(noSupp$enjQuestion, na.rm=TRUE)), colour="black", size=1)+ xlim(-20 , 20)+ylim(0,0.25)
```



The data in the histograms shows a deviation from normal distribution. Therefore, Non parametric tests was used.

Kruskal-Wallis test was used to uncover the general differences between the two groups.

```
kruskal.test(P_data$enjQuestion ~ Group, data=P_data)
```

Table 5: Kruskal-Wallis rank sum test: `P_data$enjQuestion` by `Group`

Test statistic	df	P value
3.705	2	0.1569

The p-value is 0.1569, which is greater than 0.05.

5.1.2 Enjoyment Questionnaire

checking for reliability

```
relUPost<-data.frame(P_data$Enjoy1, P_data$Enjoy2, P_data$Enjoy3, P_data$Enjoy4, P_data$Enjoy5, P_data$
alpha(relUPost)
```

```
##
```

```
## Reliability analysis
## Call: alpha(x = relUPost)
##
##   raw_alpha std.alpha G6(smc) average_r S/N    ase mean  sd median_r
##       0.95      0.95    0.98      0.72  18 0.0061    2 1.5     0.89
##
##  lower alpha upper     95% confidence boundaries
## 0.94 0.95 0.96
##
##  Reliability if an item is dropped:
##               raw_alpha std.alpha G6(smc) average_r S/N alpha se var.r med.r
## P_data.Enjoy1      0.93      0.93    0.98      0.68  13   0.0087 0.062  0.48
## P_data.Enjoy2      0.93      0.93    0.98      0.69  13   0.0085 0.061  0.51
## P_data.Enjoy3      0.96      0.96    0.97      0.79  23   0.0041 0.053  0.94
## P_data.Enjoy4      0.96      0.96    0.97      0.79  22   0.0048 0.058  0.94
## P_data.Enjoy5      0.93      0.93    0.98      0.69  14   0.0083 0.061  0.51
## P_data.Enjoy6      0.93      0.93    0.97      0.69  13   0.0084 0.058  0.51
## P_data.Enjoy7      0.94      0.93    0.98      0.69  14   0.0082 0.062  0.51
##
##  Item statistics
##                 n raw.r std.r r.cor r.drop mean  sd
## P_data.Enjoy1 174  0.97  0.96  0.96   0.96  2.3 1.8
## P_data.Enjoy2 174  0.95  0.94  0.94   0.93  2.3 1.9
## P_data.Enjoy3 174  0.65  0.69  0.66   0.56  1.7 1.6
## P_data.Enjoy4 174  0.67  0.70  0.68   0.59  1.5 1.4
## P_data.Enjoy5 174  0.95  0.93  0.93   0.92  2.2 1.8
## P_data.Enjoy6 174  0.96  0.94  0.95   0.93  2.3 1.8
## P_data.Enjoy7 174  0.94  0.93  0.92   0.92  2.0 1.7
##
## Non missing response frequency for each item
##                  0    1    2    3    4    5 miss
## P_data.Enjoy1 0.33 0.03 0.10 0.10 0.39 0.05    0
## P_data.Enjoy2 0.33 0.06 0.09 0.14 0.26 0.11    0
## P_data.Enjoy3 0.33 0.12 0.32 0.06 0.11 0.06    0
## P_data.Enjoy4 0.33 0.18 0.30 0.07 0.10 0.02    0
## P_data.Enjoy5 0.33 0.06 0.09 0.15 0.28 0.09    0
## P_data.Enjoy6 0.33 0.06 0.07 0.16 0.30 0.08    0
## P_data.Enjoy7 0.33 0.09 0.14 0.18 0.23 0.03    0
```

The alpha is > 0.7, which is acceptable. To Prepare the data, we got the average of the answers, and eliminated the no support group from analysis.

```r
P_data$EnjoymentIndex = (P_data$Enjoy1 + P_data$Enjoy2 + P_data$Enjoy3 + P_data$Enjoy4 + P_data$Enjoy5


chatbot= P_data[P_data$Group=="Chatbot Support", ]
txtSupp = P_data[P_data$Group=="Text Support", ]

EnjIndex <- rbind(chatbot, txtSupp)
```

To check normality:

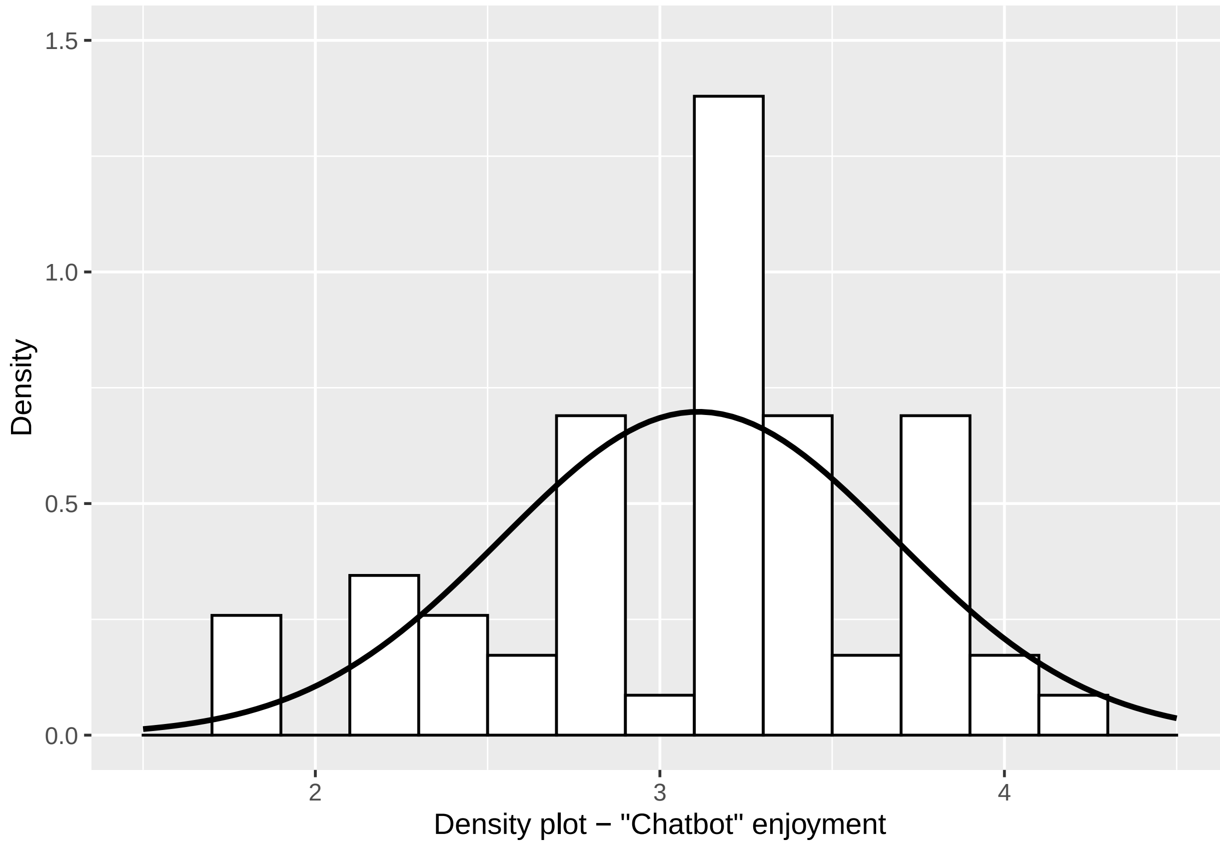
```r
stem(chatbot$EnjoymentIndex)
```

```
##
##   The decimal point is 1 digit(s) to the left of the |
##
##   18 | 666
##   20 | 44
##   22 | 99
##   24 | 33377
##   26 | 1111
##   28 | 6666
##   30 | 0444444444
##   32 | 9999999
##   34 | 3333333377
##   36 | 1111
##   38 | 6666
##   40 | 004
```

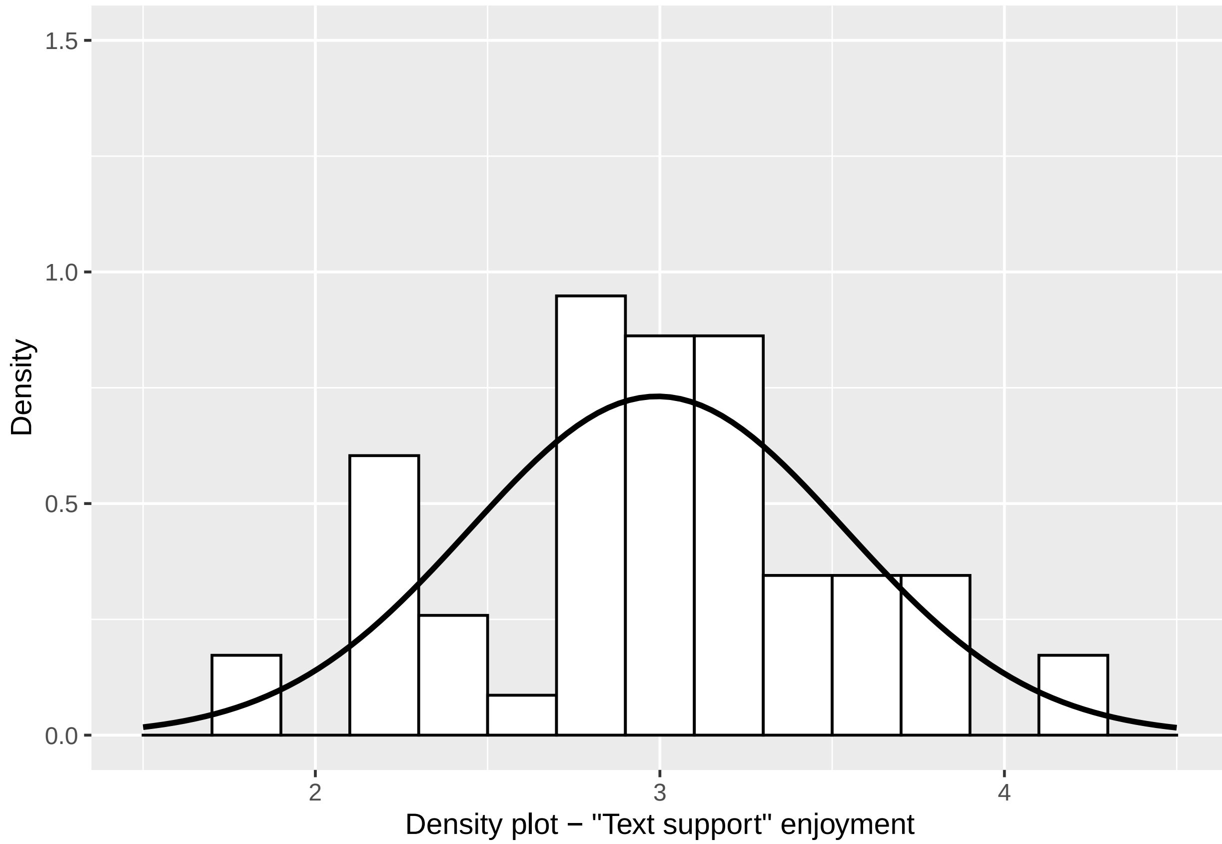
```r
stem(txtSupp$EnjoymentIndex)
```

```
##
##   The decimal point is at the |
##
##   1 | 79
##   2 | 1113333444
##   2 | 677779999999
##   3 | 0000000000111113333334444
##   3 | 66667779
##   4 | 33
```

```r
ggplot(chatbot,
aes(EnjoymentIndex)) + geom_histogram(aes(y=..density..),binwidth = 0.2,
colour="black", fill="white") + labs(x='Density plot - "Chatbot" enjoyment',
y="Density") + stat_function(fun=dnorm,
args=list(mean=mean(chatbot$EnjoymentIndex, na.rm=TRUE),
sd=sd(chatbot$EnjoymentIndex, na.rm=TRUE)), colour="black", size=1)+ xlim(1.5 , 4.5)+ ylim(0,1.5)
```




```
ggplot(txtSupp,
aes(EnjoymentIndex)) + geom_histogram(aes(y=..density..),binwidth = 0.2,
colour="black", fill="white") + labs(x='Density plot - "Text support" enjoyment',
y="Density") + stat_function(fun=dnorm,
args=list(mean=mean(txtSupp$EnjoymentIndex, na.rm=TRUE),
sd=sd(txtSupp$EnjoymentIndex, na.rm=TRUE)), colour="black", size=1)+ xlim(1.5 , 4.5)+ ylim(0,1.5)
```



The data in the histograms shows a deviation from normal distribution. Therefore, Non parametric tests was used.

Kruskal-Wallis test was used to uncover the general differences between the two groups.

```
kruskal.test(EnjIndex$EnjoymentIndex ~ Group, data=EnjIndex)
```

Table 6: Kruskal-Wallis rank sum test: `EnjIndex$EnjoymentIndex` by `Group`

Test statistic	df	P value
2.107	1	0.1466

The p-value is 0.1466, which is greater than 0.05.

5.2 Engagement

For engagement, we will check the total number of written arguments and the engagement time.

```
P_data$numberArgForNeg <- str_count(P_data$ArgForNeg, "\\\\n")
P_data$numberArgAgainstNeg <- str_count(P_data$ArgAgainstNeg, "\\\\n")
P_data$numberArgForPos <- str_count(P_data$ArgForPos, "\\\\n")
P_data$numberArgAgainstPos <- str_count(P_data$ArgAgainstPos, "\\\\n")
```

```r
P_data$numberArgs = P_data$numberArgAgainstPos+ P_data$numberArgForPos + P_data$numberArgAgainstNeg + P_



chatbot= P_data[P_data$Group=="Chatbot Support", ]
noSupp = P_data[P_data$Group=="No Support", ]
txtSupp = P_data[P_data$Group=="Text Support", ]
```

Kruskal-Wallis test was used to uncover the general differences between the two groups.

```r
kruskal.test(P_data$`Time-Spent` ~ Group, data=P_data)
```

Table 7: Kruskal-Wallis rank sum test: P_data$Time-Spent‘byGroup‘

Test statistic	df	P value
0.7659	2	0.6818

```r
kruskal.test(P_data$numberArgs ~ Group, data=P_data)
```

Table 8: Kruskal-Wallis rank sum test: `P_data$numberArgs` by `Group`

Test statistic	df	P value
0.1251	2	0.9394

The p-values are greater than 0.05.

5.3 Barrier believability

Prepare the data by getting the delta between post and pre of the highest rated barrier.

```r
#barrier believability & thought believability
BarrTemp = P_data

BarrNC <- BarrTemp[BarrTemp$HighestBarrier=="Not Convinced", ]
BarrLE <- BarrTemp[BarrTemp$HighestBarrier=="Low self-efficacy", ]
BarrPerf <- BarrTemp[BarrTemp$HighestBarrier=="Perfectionism", ]

BarrNC$BarrDelta= BarrNC$`NotConvnced-Pre` - BarrNC$`NotConvnced-Post`
BarrLE$BarrDelta= BarrLE$`LowEff-Pre` - BarrLE$`LowEff-Post`
BarrPerf$BarrDelta= BarrPerf$`Perf-Pre`- BarrPerf$`Perf-Post`



Barr <- rbind(BarrNC, BarrLE, BarrPerf)
```

```r
chatbotB= Barr[Barr$Group=="Chatbot Support", ]
noSuppB = Barr[Barr$Group=="No Support", ]
txtSuppB = Barr[Barr$Group=="Text Support", ]
```

checking for normality

```r
stem(chatbotB$BarrDelta)
```

```
##
##   The decimal point is 1 digit(s) to the right of the |
##
##    -8 | 0
##    -6 |
##    -4 |
##    -2 | 050
##    -0 | 0
##     0 | 0000000000255555555550000000004559
##     2 | 000000050000005
##     4 | 058
##     6 |
##     8 | 0
```

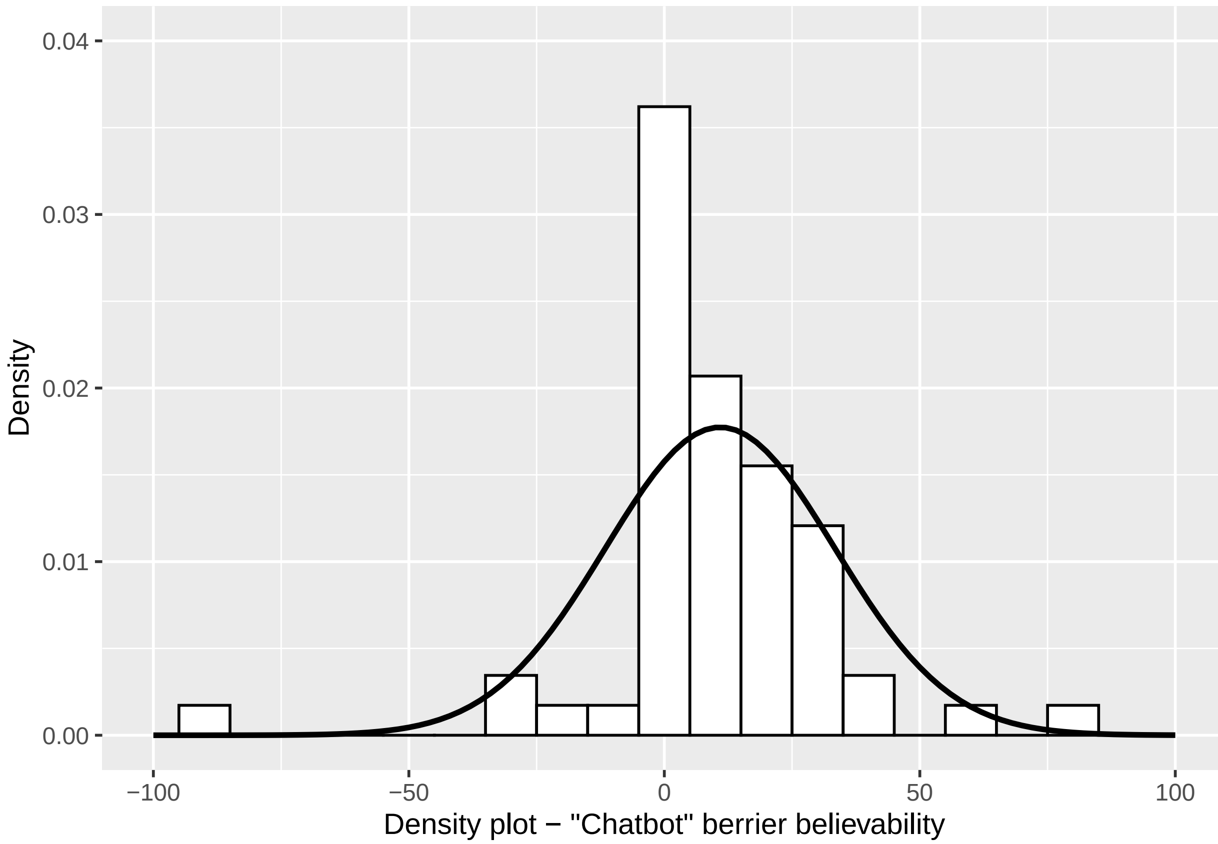
```r
stem(txtSuppB$BarrDelta)
```

```
##
##   The decimal point is 1 digit(s) to the right of the |
##
##   -8 | 0
##   -6 |
##   -4 |
##   -2 | 0
##   -0 | 0000055
##    0 | 0000000000055550000000055
##    2 | 0000005555600255
##    4 | 0000010
##    6 | 5
##    8 | 0
```

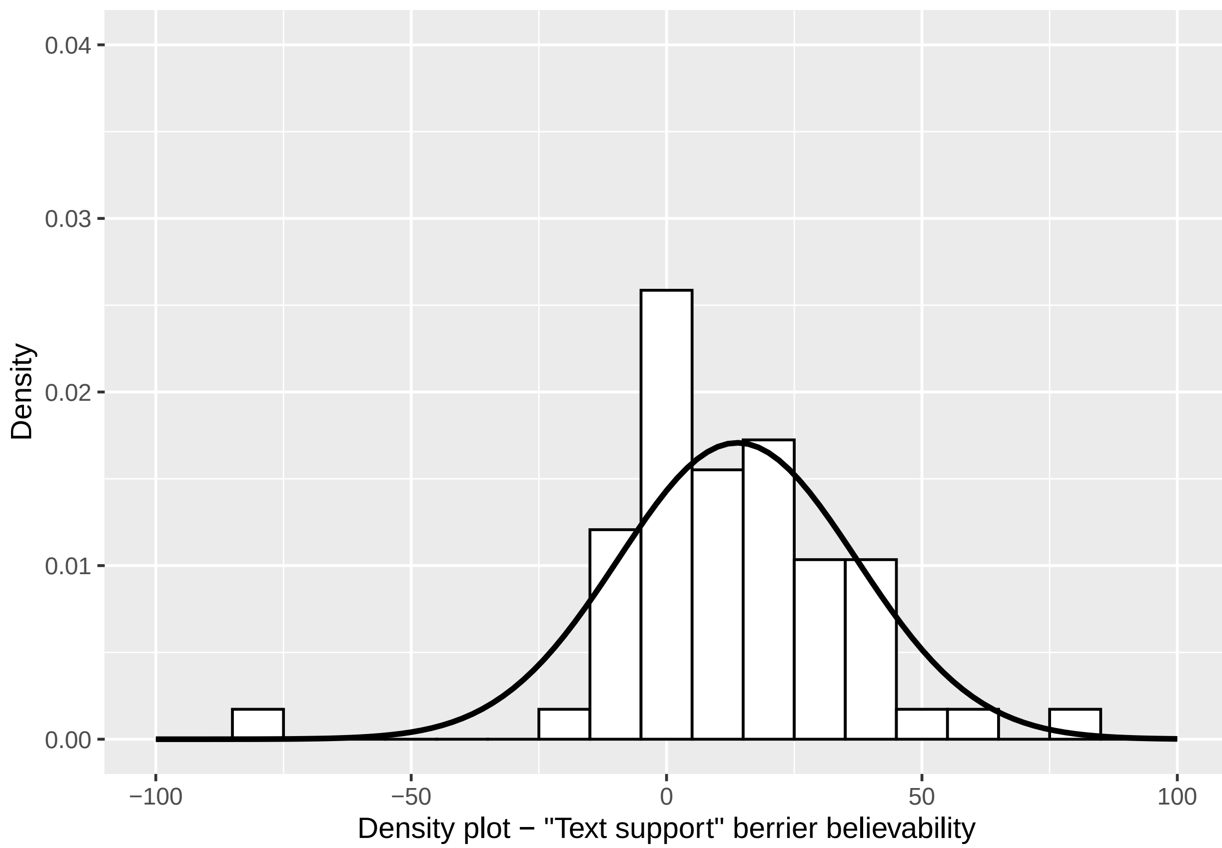
```r
stem(noSuppB$BarrDelta)
```

```
##
##   The decimal point is 1 digit(s) to the right of the |
##
##   -2 | 50
##   -0 | 950055
##    0 | 0000000000055555555000000055
##    2 | 00000055550005
##    4 | 0005505
##    6 | 5
##    8 | 2
```

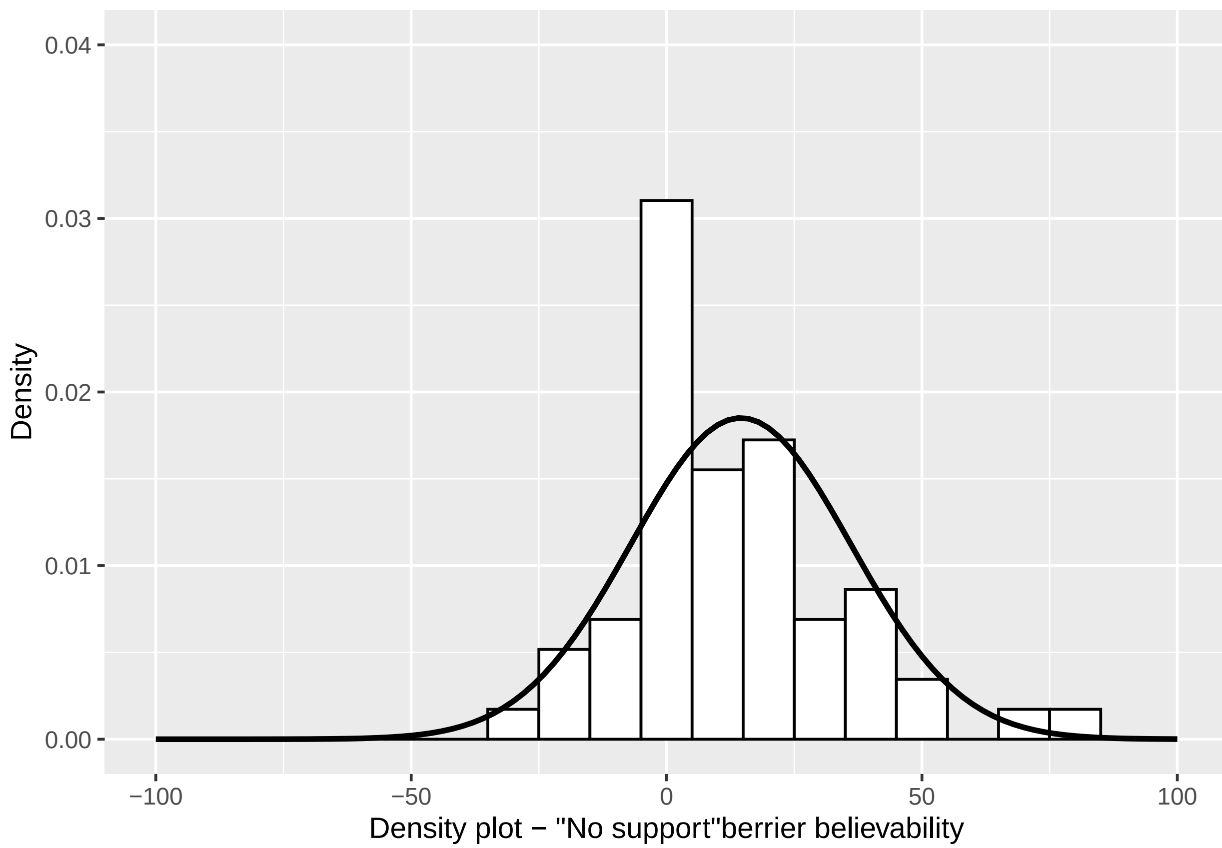
```
ggplot(chatbotB,
aes(BarrDelta)) + geom_histogram(aes(y=..density..),binwidth = 10,
colour="black", fill="white") + labs(x='Density plot - "Chatbot" berrier believability',
y="Density") + stat_function(fun=dnorm,
args=list(mean=mean(chatbotB$BarrDelta, na.rm=TRUE),
sd=sd(chatbotB$BarrDelta, na.rm=TRUE)), colour="black", size=1)+ xlim(-100 , 100)+ ylim(0,0.04)
```




```
ggplot(txtSuppB,
aes(BarrDelta)) + geom_histogram(aes(y=..density..),binwidth = 10,
colour="black", fill="white") + labs(x='Density plot - "Text support" berrier believability',
y="Density") + stat_function(fun=dnorm,
args=list(mean=mean(txtSuppB$BarrDelta, na.rm=TRUE),
sd=sd(txtSuppB$BarrDelta, na.rm=TRUE)), colour="black", size=1)+ xlim(-100 , 100)+ ylim(0,0.04)
```




```
ggplot(noSuppB,
aes(BarrDelta)) + geom_histogram(aes(y=..density..),binwidth = 10,
colour="black", fill="white") + labs(x='Density plot - "No support"berrier believability',
y="Density") + stat_function(fun=dnorm,
args=list(mean=mean(noSuppB$BarrDelta, na.rm=TRUE),
sd=sd(noSuppB$BarrDelta, na.rm=TRUE)), colour="black", size=1)+ xlim(-100 , 100)+ ylim(0,0.04)
```



The data in the histograms shows a deviation from normal distribution. Therefore, Non parametric tests was used.

Kruskal-Wallis test was used to uncover the general differences between the three groups.

```
kruskal.test(BarrDelta ~ Group, data=Barr)
```

Table 9: Kruskal-Wallis rank sum test: `BarrDelta` by `Group`

Test statistic	df	P value
0.189	2	0.9098

The p-value is greater than 0.05, so we can not reject the null hypothesis.

5.4 Thoughts believability

Prepare the data by getting the delta between post and pre of the highest rated negative and positive thoughts.

```
ThTemp = P_data

Th1 <- ThTemp[ThTemp$HighestNegThou=="TrNegSce1" | ThTemp$HighestNegThou=="EmNegSce1", ]
Th2 <- ThTemp[ThTemp$HighestNegThou=="TrNegSce2" | ThTemp$HighestNegThou=="EmNegSce2", ]
Th3 <- ThTemp[ThTemp$HighestNegThou=="TrNegSce3" | ThTemp$HighestNegThou=="EmNegSce3", ]
```

```r
Th1$thNegDelta = Th1$PreNegTh1 - Th1$PostNegTh1
Th2$thNegDelta = Th2$PreNegTh2 - Th2$PostNegTh2
Th3$thNegDelta = Th3$PreNegTh3 - Th3$PostNegTh3

Th1$thPosDelta = Th1$PrePosTh1 - Th1$PostPosTh1
Th2$thPosDelta = Th2$PrePosTh2 - Th2$PostPosTh2
Th3$thPosDelta = Th3$PrePosTh3 - Th3$PostPosTh3



ThBind <- rbind(Th1, Th2, Th3)

Tho = ThBind %>% select(ParticipantID, Group, HighestNegThou, thNegDelta, thPosDelta)
Tho$TIndex= Tho$thPosDelta + Tho$thNegDelta

chatbotT= Tho[Tho$Group=="Chatbot Support", ]
noSuppT = Tho[Tho$Group=="No Support", ]
txtSuppT = Tho[Tho$Group=="Text Support", ]
```

Normality checking:

```r
stem(chatbotT$TIndex)
```

```
##
##   The decimal point is 1 digit(s) to the right of the |
##
##   -8 | 0
##   -6 |
##   -4 |
##   -2 | 050
##   -0 | 50000055555
##    0 | 0000000004555558000055
##    2 | 00005500000
##    4 | 000000
##    6 | 50
##    8 | 95
```

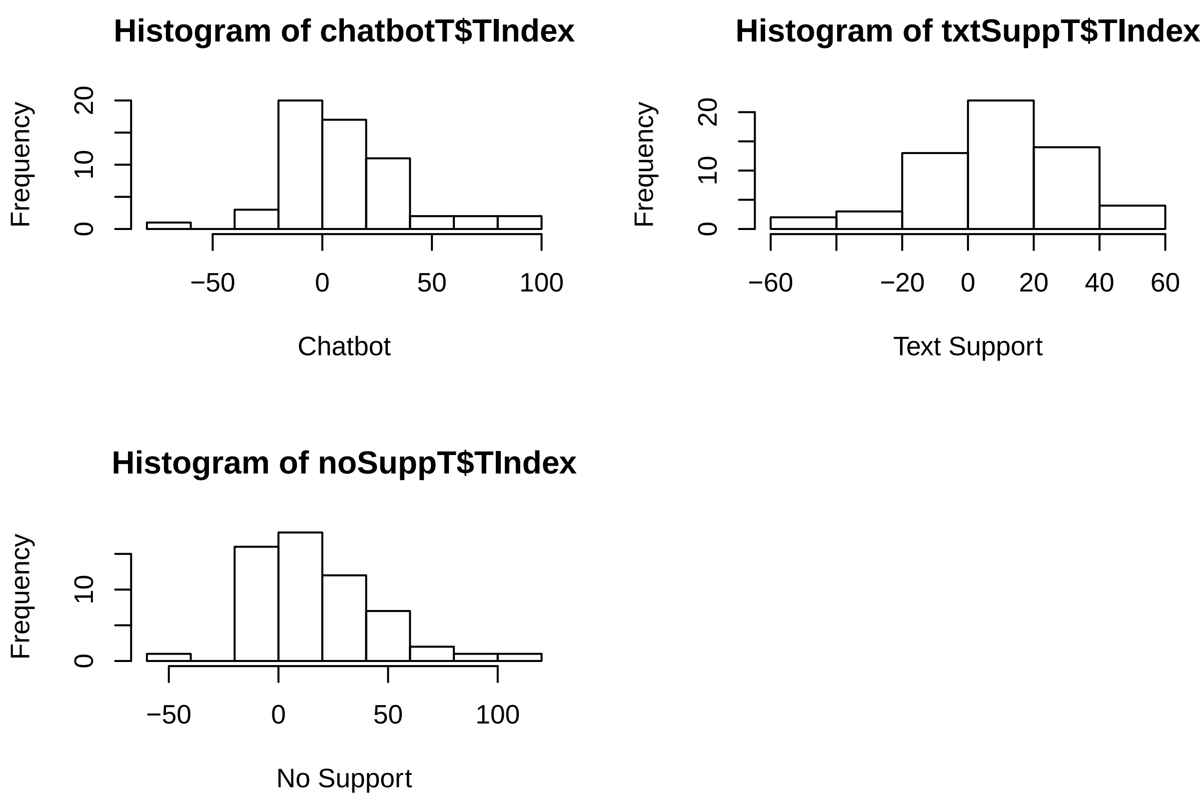
```r
stem(txtSuppT$TIndex)
```

```
##
##   The decimal point is 1 digit(s) to the right of the |
##
##   -6 | 0
##   -4 | 0
##   -2 | 000
##   -0 | 555503
##    0 | 00000001555559900000555559
##    2 | 00005555500000555
##    4 | 00055
```

```r
stem(noSuppT$TIndex)
```

```
##
##   The decimal point is 1 digit(s) to the right of the |
##
##   -4 | 5
##   -2 |
##   -0 | 5000555
##    0 | 0000000023555900002259
##    2 | 0000580000055
##    4 | 0005550000
##    6 | 50
##    8 | 0
##   10 | 0
```

```r
attach(mtcars)
par(mfrow=c(2,2))
hist(chatbotT$TIndex, xlab="Chatbot")
hist(txtSuppT$TIndex, xlab="Text Support")
hist(noSuppT$TIndex, xlab="No Support")
```



The data in the histograms shows a deviation from normal distribution. Therefore, Non parametric tests was used.

Kruskal-Wallis test was used to uncover the general differences between the three groups.

```r
kruskal.test(TIndex ~ Group, data=Tho)
```

Table 10: Kruskal-Wallis rank sum test: `TIndex` by `Group`

Test statistic	df	P value
3.19	2	0.203

The p-value is greater than 0.05, so we can not reject the null hypothesis.