

CircleTime Data Analyses

Pourya Shahverdi

2023-11-22

Contents

1	About the Study	2
1.1	Introduction to Circle-Time	2
1.2	Research Questions	2
1.3	Study Design	2
	1.3.0.1 Independent Variables	2
	1.3.0.2 Dependent Variables	3
1.3.1	Data Collection	3
	1.3.1.1 Affect	3
	1.3.1.2 Communication	3
	1.3.1.3 Engagement	3
	1.3.1.4 Performance	3
	1.3.1.5 Inter-observer Agreement (IoA)	4
2	Data Analysis	4
2.1	Affect	4
	2.1.1 Affect Positive	4
	2.1.2 Affect Negative	7
	2.1.3 Affect Neutral	9
2.2	Communication	12
	2.2.1 Communication with Instructor	12
	2.2.2 Communication with Instructor Prompted	15
	2.2.3 Communication with Therapist	18
	2.2.4 Communication with Indeterminate	21
2.3	Engagement	24
	2.3.1 Engagement with On Target	24
	2.3.2 Engagement with Therapist	27
	2.3.3 Engagement with OffTarget	30

2.4	Performance	33
2.4.1	Performance Positive	33
2.4.2	Performance Negative	36

1 About the Study

1.1 Introduction to Circle-Time

Circle-time is a group activity based on Applied Behavior Analysis (ABA) for children with Autism Spectrum Disorder (ASD) to prepare them for attending in traditional classroom activities alongside neurotically developed children. In circle-time, children sit together semicircular, and an instructor give them group instruction activities such as dance, yoga, labeling animals, finding objects, etc. The goal of circle-time is to improve children’s learning behaviors, which are:

- Affect
- Communication
- Engagement
- Performance

In this study, we evaluate the efficacy of a social robot in delivering group instruction activities to children with ASD. Throughout the six month of experiment, Six children participants received 10 sessions of group instructions from a human instructor and 10 sessions from a Pepper humanoid social robot as a within-subject study design. To compare children learning behaviors between the human and the robot instructor conditions their activities were video recorded and coded for the sessions 1, 4, 7, and 10.

1.2 Research Questions

In this research, we address the following research questions:

1. How learning behaviors (Affect, Communication, Engagement, Performance) differ between conditions (Human, Robot)?
2. How is the interaction between learning behaviors (Affect, Communication, Engagement, Performance) and time (Session Number)?
3. How is the interaction between different learning behaviors (Affect, Communication, Engagement, Performance)?
4. Between the Affect, Communication, and Engagement learning behaviors, which one is more effective on the Performance learning behavior (increasing performance is considered as the main objective)?

1.3 Study Design

For this longitudinal within-subject study with 6 participants we defined the following variables:

1.3.0.1 Independent Variables

- Instructor Conditions:
 - Human ~ 1
 - Robot ~ 2

- Time
 - Session 1 ~ 1
 - Session 4 ~ 2
 - Session 7 ~ 3
 - Session 10 ~ 4

1.3.0.2 Dependent Variables

- Affect
- Communication
- Engagement
- Performance

1.3.1 Data Collection

The evaluation of the learning behavior is based on the following metrics:

1.3.1.1 Affect children's happiness level was defined as:

- Positive
- Negative
- Neutral

A video was divided into 10 seconds intervals, and a human coder, focusing on one child in the group, labeled that interval as Positive if the child was happy, Negative if they was sad, and Neutral if they was neither happy or sad. Percentage of each measurement is used for analysis.

1.3.1.2 Communication Communication of the children was coded into 4 categories. Communication with:

- Instructor
- Instructor-Prompted
- Behavior Therapist (BT) or peers
- Indeterminate

1.3.1.3 Engagement Engagement was coded into 3 categories. Engagement with:

- Instructor or screen (On Target)
- BT or peers
- Off Target

1.3.1.4 Performance Children's performance was coded into two categories:

- Positive
- Negative

1.3.1.5 Inter-observer Agreement (IoA) At the beginning of the coding procedure, coders' understanding of the metrics had to be on the same page. We used Cohen's Kappa score to evaluate the IoA on the coding procedure. An individual coder was allowed to code independently only if their Cohen's Kappa IoA score was higher than 80%. All session ones and tens were double coded as well as the 30% of the session fours and sevens. For the sessions with lower than 80% agreement, coders went through coding together and came up with 100% agreement. We considered this conservative approach since we were looking into the highest reliability of data on our 6 participants.

2 Data Analysis

In order to investigate the research questions, we analyze the data from children's learning behaviors as follows:

2.1 Affect

We use a Linear Mixed-Effect Model to analyze the Affect metric.

2.1.1 Affect Positive

```
library(readr)
library(lme4)

## Loading required package: Matrix

CircleTimeData <- read_csv("~/GitHub/CircleTime_HRI_DataAnalysis/CircleTimeData.csv")

## Rows: 48 Columns: 15

## -- Column specification -----
## Delimiter: ","
## dbl (15): Subject, Condition, time, Affect_Positive, Affect_Negative, Affect...
##
## 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.

# Fit a mixed-effects model
df <- CircleTimeData
model <- lmer(Affect_Positive ~ Condition * time + (1 | Subject), data = df)
summary(model)

## Linear mixed model fit by REML ['lmerMod']
## Formula: Affect_Positive ~ Condition * time + (1 | Subject)
## Data: df
##
## REML criterion at convergence: -85.1
##
## Scaled residuals:
```

```
##      Min      1Q  Median      3Q      Max
## -2.0016 -0.7370 -0.0926  0.5711  1.7623
##
## Random effects:
##   Groups   Name                Variance Std.Dev.
##   Subject  (Intercept)  0.003943  0.06279
##   Residual                    0.005010  0.07078
## Number of obs: 48, groups:  Subject, 6
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)    0.044179   0.083186   0.531
## Condition      0.089989   0.050051   1.798
## time           0.008023   0.028897   0.278
## Condition:time -0.013518   0.018276  -0.740
##
## Correlation of Fixed Effects:
##              (Intr) Condtn time
## Condition    -0.903
## time         -0.868  0.866
## Conditin:tm  0.824 -0.913 -0.949
```

```
summary_table <- summary(model)$coefficients
t_values <- summary_table[, "t value"]

# Number of observations
num_obs <- nobs(model)

for (i in seq_along(t_values)) {
  # Extract t-value for the term
  t_value <- t_values[i]
  num_obs_term <- num_obs

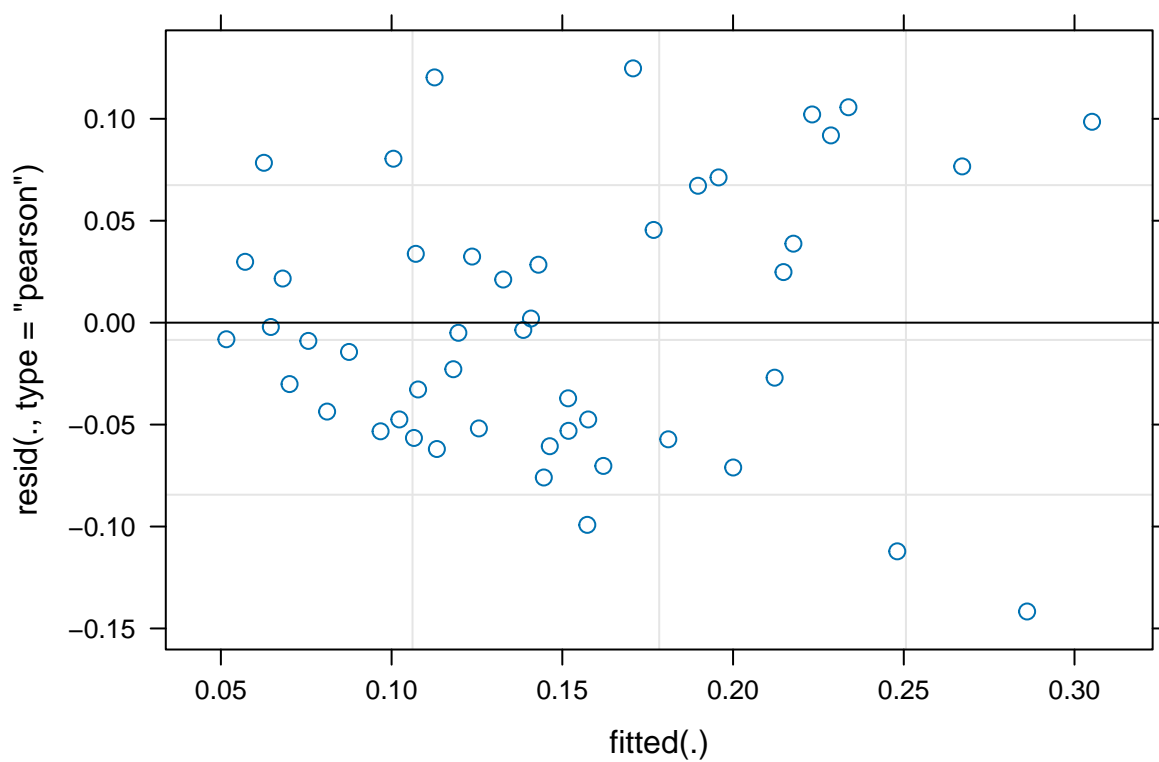
  # Calculate p-value using t-value and degrees of freedom
  df <- num_obs_term - 1 # degrees of freedom for a fixed effect
  p_value <- 2 * (1 - pt(abs(t_value), df))

  # Print or use the results as needed
  cat("Term:", names(t_values)[i], "\n")
  cat("  T-value:", t_value, "\n")
  cat("  Number of Observations:", num_obs_term, "\n")
  cat("  P-value:", p_value, "\n\n")
}
```

```
## Term: (Intercept)
##   T-value: 0.5310853
##   Number of Observations: 48
##   P-value: 0.5978607
##
## Term: Condition
##   T-value: 1.79794
##   Number of Observations: 48
##   P-value: 0.07861156
```

```
##
## Term: time
##   T-value: 0.2776264
##   Number of Observations: 48
##   P-value: 0.7825168
##
## Term: Condition:time
##   T-value: -0.739676
##   Number of Observations: 48
##   P-value: 0.4631753
```

```
plot(model)
```



```
# Shapiro-Wilk test for normality
shapiro.test(residuals(model))
```

```
##
## Shapiro-Wilk normality test
##
## data: residuals(model)
## W = 0.97212, p-value = 0.3058
```

2.1.2 Affect Negative

```
library(readr)
library(lme4)

CircleTimeData <- read_csv("~/GitHub/CircleTime_HRI_DataAnalysis/CircleTimeData.csv")

## Rows: 48 Columns: 15
## -- Column specification -----
## Delimiter: ","
## dbl (15): Subject, Condition, time, Affect_Positive, Affect_Negative, Affect...
##
## 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.

# View(CircleTimeData)
# head(CircleTimeData)

# Fit a mixed-effects model

df <- CircleTimeData
# spec(df)
model <- lmer(Affect_Negative ~ Condition * time + (1 | Subject), data = df)
summary(model)

## Linear mixed model fit by REML ['lmerMod']
## Formula: Affect_Negative ~ Condition * time + (1 | Subject)
## Data: df
##
## REML criterion at convergence: -213.9
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.84657 -0.44977 -0.21072  0.03255  2.76443
##
## Random effects:
##  Groups   Name                Variance Std.Dev.
## Subject (Intercept) 8.637e-05 0.009294
## Residual          2.931e-04 0.017120
## Number of obs: 48, groups: Subject, 6
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)    0.021188   0.019513   1.086
## Condition     -0.009739   0.012106  -0.804
## time           -0.006677   0.006989  -0.955
## Condition:time  0.005215   0.004420   1.180
##
## Correlation of Fixed Effects:
##              (Intr) Condtn time
## Condition    -0.931
## time         -0.895  0.866
## Conditin:tm  0.849 -0.913 -0.949
```

```

summary_table <- summary(model)$coefficients
t_values <- summary_table[, "t value"]

# Number of observations
num_obs <- nobs(model)

for (i in seq_along(t_values)) {
  # Extract t-value for the term
  t_value <- t_values[i]
  num_obs_term <- num_obs

  # Calculate p-value using t-value and degrees of freedom
  df <- num_obs_term - 1 # degrees of freedom for a fixed effect
  p_value <- 2 * (1 - pt(abs(t_value), df))

  # Print or use the results as needed
  cat("Term:", names(t_values)[i], "\n")
  cat("  T-value:", t_value, "\n")
  cat("  Number of Observations:", num_obs_term, "\n")
  cat("  P-value:", p_value, "\n\n")
}

```

```

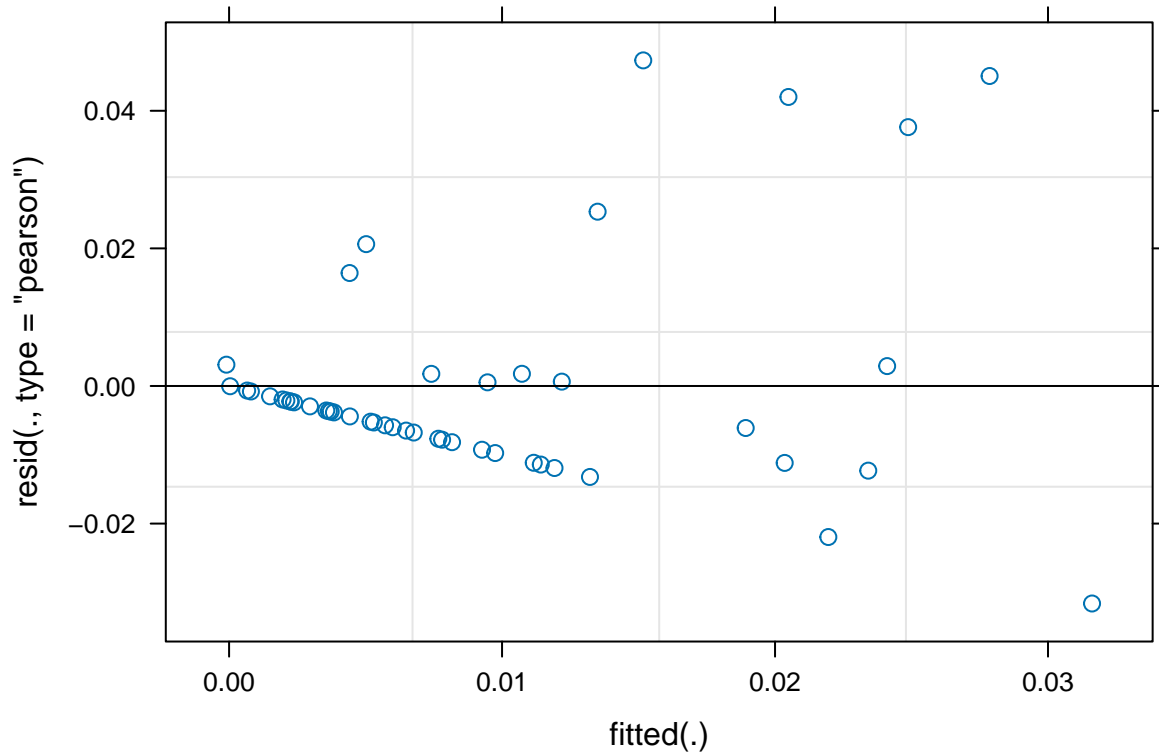
## Term: (Intercept)
##   T-value: 1.085835
##   Number of Observations: 48
##   P-value: 0.2830901
##
## Term: Condition
##   T-value: -0.8044797
##   Number of Observations: 48
##   P-value: 0.4251707
##
## Term: time
##   T-value: -0.9553375
##   Number of Observations: 48
##   P-value: 0.3442952
##
## Term: Condition:time
##   T-value: 1.179807
##   Number of Observations: 48
##   P-value: 0.2440142

```

```

plot(model)

```

```
# Shapiro-Wilk test for normality
shapiro.test(residuals(model))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(model)
## W = 0.78183, p-value = 5.167e-07
```

2.1.3 Affect Neutral

```
library(readr)
library(lme4)

CircleTimeData <- read_csv("~/GitHub/CircleTime_HRI_DataAnalysis/CircleTimeData.csv")
```

```
## Rows: 48 Columns: 15
## -- Column specification -----
## Delimiter: ","
## dbl (15): Subject, Condition, time, Affect_Positive, Affect_Negative, Affect...
##
## 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.
```

```

# View(CircleTimeData)
# head(CircleTimeData)

# Fit a mixed-effects model

df <- CircleTimeData
# spec(df)
model <- lmer(Affect_Neutral ~ Condition * time + (1 | Subject), data = df)
summary(model)

```

```

## Linear mixed model fit by REML ['lmerMod']
## Formula: Affect_Neutral ~ Condition * time + (1 | Subject)
## Data: df
##
## REML criterion at convergence: -85.8
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.66540 -0.57237  0.02484  0.79473  1.99871
##
## Random effects:
## Groups Name Variance Std.Dev.
## Subject (Intercept) 0.003314 0.05757
## Residual 0.005007 0.07076
## Number of obs: 48, groups: Subject, 6
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)  0.949397  0.082533  11.503
## Condition   -0.088887  0.050037  -1.776
## time        -0.007811  0.028889  -0.270
## Condition:time 0.012292  0.018271  0.673
##
## Correlation of Fixed Effects:
##              (Intr) Condtn time
## Condition   -0.909
## time        -0.875  0.866
## Conditin:tm  0.830 -0.913 -0.949

```

```

summary_table <- summary(model)$coefficients
t_values <- summary_table[, "t value"]

# Number of observations
num_obs <- nobs(model)

for (i in seq_along(t_values)) {
  # Extract t-value for the term
  t_value <- t_values[i]
  num_obs_term <- num_obs

  # Calculate p-value using t-value and degrees of freedom
  df <- num_obs_term - 1 # degrees of freedom for a fixed effect

```

```

p_value <- 2 * (1 - pt(abs(t_value), df))

# Print or use the results as needed
cat("Term:", names(t_values)[i], "\n")
cat("  T-value:", t_value, "\n")
cat("  Number of Observations:", num_obs_term, "\n")
cat("  P-value:", p_value, "\n\n")
}

```

```

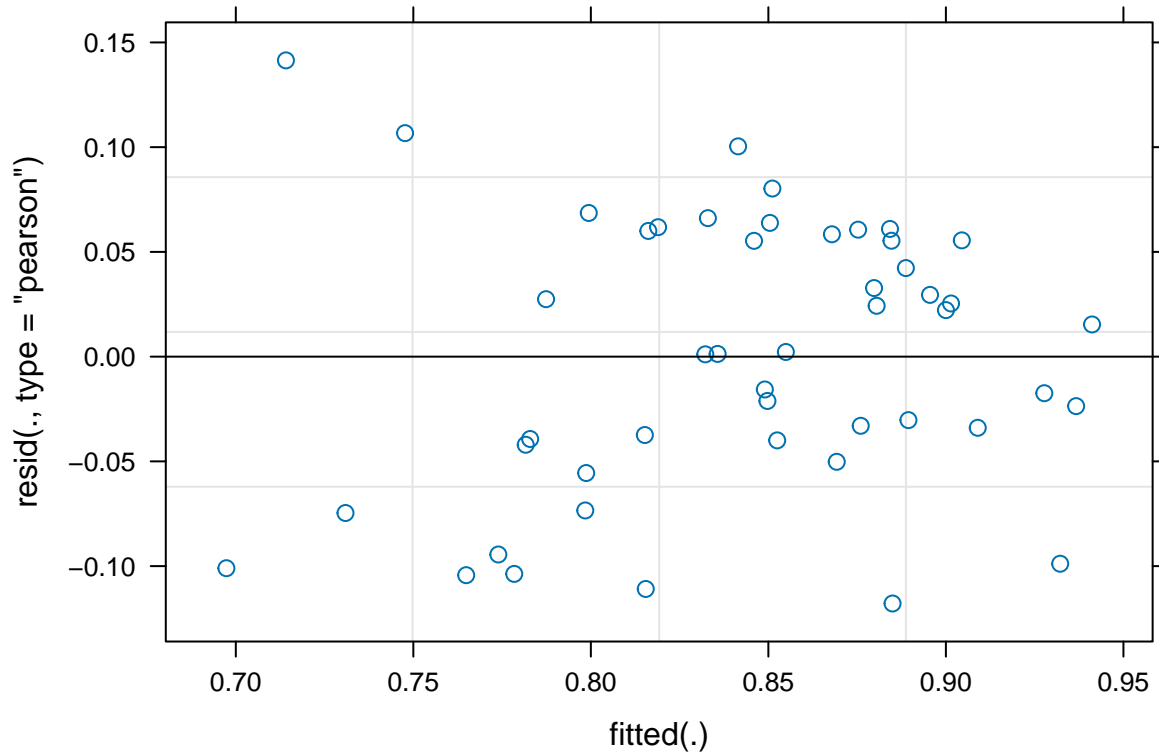
## Term: (Intercept)
##   T-value: 11.50324
##   Number of Observations: 48
##   P-value: 2.88658e-15
##
## Term: Condition
##   T-value: -1.776409
##   Number of Observations: 48
##   P-value: 0.08213864
##
## Term: time
##   T-value: -0.2703727
##   Number of Observations: 48
##   P-value: 0.7880573
##
## Term: Condition:time
##   T-value: 0.6727798
##   Number of Observations: 48
##   P-value: 0.504382

```

```

plot(model)

```



```
# Shapiro-Wilk test for normality
shapiro.test(residuals(model))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(model)
## W = 0.96467, p-value = 0.1558
```

2.2 Communication

2.2.1 Communication with Instructor

```
library(readr)
library(lme4)

CircleTimeData <- read_csv("~/GitHub/CircleTime_HRI_DataAnalysis/CircleTimeData.csv")

## Rows: 48 Columns: 15
## -- Column specification -----
## Delimiter: ","
## dbl (15): Subject, Condition, time, Affect_Positive, Affect_Negative, Affect...
##
```

```

## 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.

# View(CircleTimeData)
# head(CircleTimeData)

# Fit a mixed-effects model

df <- CircleTimeData
# spec(df)
model <- lmer(Communication_with_Instructor ~ Condition * time + (1 | Subject), data = df)
summary(model)

## Linear mixed model fit by REML ['lmerMod']
## Formula: Communication_with_Instructor ~ Condition * time + (1 | Subject)
## Data: df
##
## REML criterion at convergence: -119.4
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.4514 -0.5872 -0.2510  0.4615  4.2177
##
## Random effects:
## Groups Name Variance Std.Dev.
## Subject (Intercept) 0.0006867 0.02620
## Residual 0.0025256 0.05026
## Number of obs: 48, groups: Subject, 6
##
## Fixed effects:
## Estimate Std. Error t value
## (Intercept) 0.06080 0.05720 1.063
## Condition 0.04408 0.03554 1.241
## time 0.02565 0.02052 1.250
## Condition:time -0.02233 0.01298 -1.721
##
## Correlation of Fixed Effects:
## (Intr) Condtm time
## Condition -0.932
## time -0.897 0.866
## Conditin:tm 0.851 -0.913 -0.949

summary_table <- summary(model)$coefficients
t_values <- summary_table[, "t value"]

# Number of observations
num_obs <- nobs(model)

for (i in seq_along(t_values)) {
  # Extract t-value for the term
  t_value <- t_values[i]
  num_obs_term <- num_obs

```

```

# Calculate p-value using t-value and degrees of freedom
df <- num_obs_term - 1 # degrees of freedom for a fixed effect
p_value <- 2 * (1 - pt(abs(t_value), df))

# Print or use the results as needed
cat("Term:", names(t_values)[i], "\n")
cat("  T-value:", t_value, "\n")
cat("  Number of Observations:", num_obs_term, "\n")
cat("  P-value:", p_value, "\n\n")
}

```

```

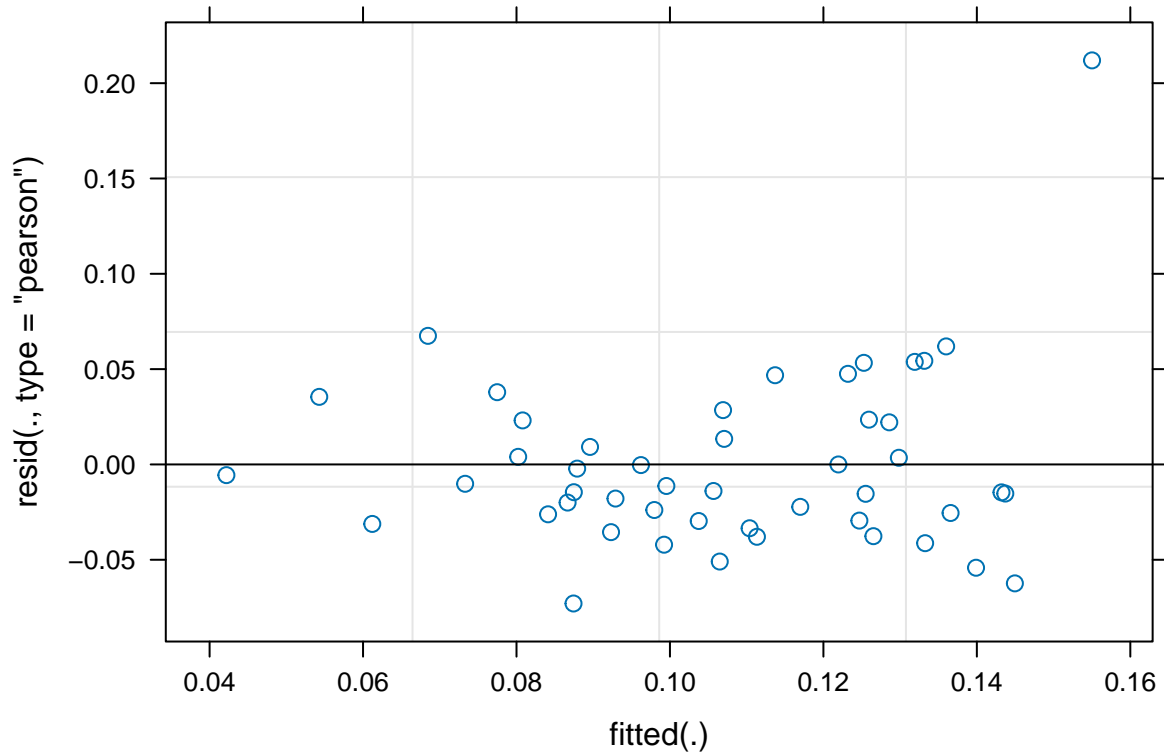
## Term: (Intercept)
##   T-value: 1.06307
##   Number of Observations: 48
##   P-value: 0.2931817
##
## Term: Condition
##   T-value: 1.240565
##   Number of Observations: 48
##   P-value: 0.2209239
##
## Term: time
##   T-value: 1.250138
##   Number of Observations: 48
##   P-value: 0.2174381
##
## Term: Condition:time
##   T-value: -1.72103
##   Number of Observations: 48
##   P-value: 0.09182297

```

```

plot(model)

```



```
# Shapiro-Wilk test for normality
shapiro.test(residuals(model))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(model)
## W = 0.84633, p-value = 1.74e-05
```

2.2.2 Communication with Instructor Prompted

```
library(readr)
library(lme4)

CircleTimeData <- read_csv("~/GitHub/CircleTime_HRI_DataAnalysis/CircleTimeData.csv")
```

```
## Rows: 48 Columns: 15
## -- Column specification -----
## Delimiter: ","
## dbl (15): Subject, Condition, time, Affect_Positive, Affect_Negative, Affect...
##
## 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.
```

```

# View(CircleTimeData)
# head(CircleTimeData)

# Fit a mixed-effects model

df <- CircleTimeData
# spec(df)
model <- lmer(Communication_with_Instructor_Prompted ~ Condition * time + (1 | Subject), data = df)

```

```
## boundary (singular) fit: see help('isSingular')
```

```
summary(model)
```

```

## Linear mixed model fit by REML ['lmerMod']
## Formula: Communication_with_Instructor_Prompted ~ Condition * time + (1 |
##      Subject)
##      Data: df
##
## REML criterion at convergence: -244
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.0033 -0.7118 -0.2850  0.3681  2.9508
##
## Random effects:
##      Groups   Name      Variance Std.Dev.
##      Subject (Intercept) 0.0000000 0.00000
##      Residual              0.0001694 0.01302
## Number of obs: 48, groups: Subject, 6
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)    0.008726   0.014552   0.600
## Condition      0.004001   0.009203   0.435
## time           0.001919   0.005314   0.361
## Condition:time -0.002795   0.003361  -0.832
##
## Correlation of Fixed Effects:
##              (Intr) Condtn time
## Condition   -0.949
## time        -0.913  0.866
## Condition:tm  0.866 -0.913 -0.949
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')

```

```

summary_table <- summary(model)$coefficients
t_values <- summary_table[, "t value"]

# Number of observations
num_obs <- nobs(model)

```



```

for (i in seq_along(t_values)) {
  # Extract t-value for the term
  t_value <- t_values[i]
  num_obs_term <- num_obs

  # Calculate p-value using t-value and degrees of freedom
  df <- num_obs_term - 1 # degrees of freedom for a fixed effect
  p_value <- 2 * (1 - pt(abs(t_value), df))

  # Print or use the results as needed
  cat("Term:", names(t_values)[i], "\n")
  cat("  T-value:", t_value, "\n")
  cat("  Number of Observations:", num_obs_term, "\n")
  cat("  P-value:", p_value, "\n\n")
}

```

```

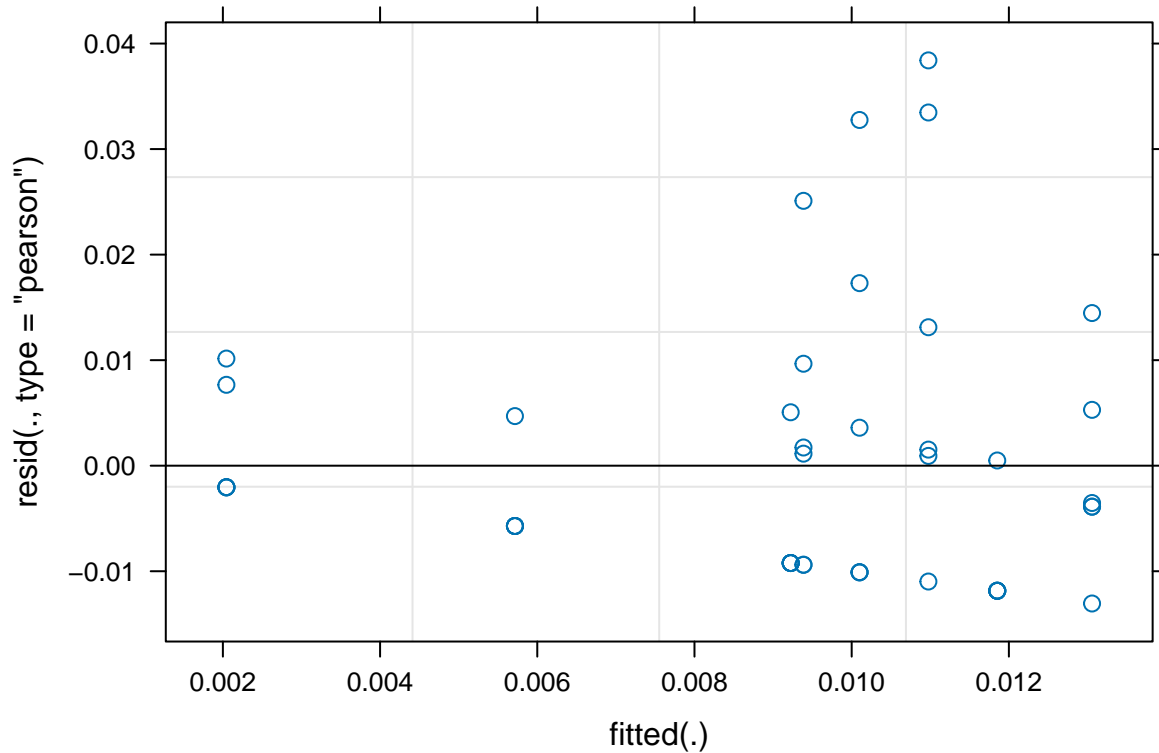
## Term: (Intercept)
##   T-value: 0.5996721
##   Number of Observations: 48
##   P-value: 0.551605
##
## Term: Condition
##   T-value: 0.4347852
##   Number of Observations: 48
##   P-value: 0.6657082
##
## Term: time
##   T-value: 0.3612166
##   Number of Observations: 48
##   P-value: 0.7195562
##
## Term: Condition:time
##   T-value: -0.8317771
##   Number of Observations: 48
##   P-value: 0.409741

```

```

plot(model)

```



```
# Shapiro-Wilk test for normality
shapiro.test(residuals(model))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(model)
## W = 0.83224, p-value = 7.584e-06
```

2.2.3 Communication with Therapist

```
library(readr)
library(lme4)

CircleTimeData <- read_csv("~/GitHub/CircleTime_HRI_DataAnalysis/CircleTimeData.csv")

## Rows: 48 Columns: 15
## -- Column specification -----
## Delimiter: ","
## dbl (15): Subject, Condition, time, Affect_Positive, Affect_Negative, Affect...
##
## 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.
```

```

# View(CircleTimeData)
# head(CircleTimeData)

# Fit a mixed-effects model

df <- CircleTimeData
# spec(df)
model <- lmer(Communication_with_Therapist ~ Condition * time + (1 | Subject), data = df)
summary(model)

```

```

## Linear mixed model fit by REML ['lmerMod']
## Formula: Communication_with_Therapist ~ Condition * time + (1 | Subject)
## Data: df
##
## REML criterion at convergence: -114.5
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.4254 -0.5918 -0.2273  0.3879  2.6041
##
## Random effects:
## Groups Name Variance Std.Dev.
## Subject (Intercept) 0.0002795 0.01672
## Residual 0.0030242 0.05499
## Number of obs: 48, groups: Subject, 6
##
## Fixed effects:
## Estimate Std. Error t value
## (Intercept) 0.159495 0.061862 2.578
## Condition -0.024849 0.038886 -0.639
## time -0.034171 0.022451 -1.522
## Condition:time 0.009994 0.014199 0.704
##
## Correlation of Fixed Effects:
## (Intr) Condtm time
## Condition -0.943
## time -0.907 0.866
## Conditin:tm 0.861 -0.913 -0.949

```

```

summary_table <- summary(model)$coefficients
t_values <- summary_table[, "t value"]

# Number of observations
num_obs <- nobs(model)

for (i in seq_along(t_values)) {
  # Extract t-value for the term
  t_value <- t_values[i]
  num_obs_term <- num_obs

  # Calculate p-value using t-value and degrees of freedom
  df <- num_obs_term - 1 # degrees of freedom for a fixed effect

```

```

p_value <- 2 * (1 - pt(abs(t_value), df))

# Print or use the results as needed
cat("Term:", names(t_values)[i], "\n")
cat("  T-value:", t_value, "\n")
cat("  Number of Observations:", num_obs_term, "\n")
cat("  P-value:", p_value, "\n\n")
}

```

```

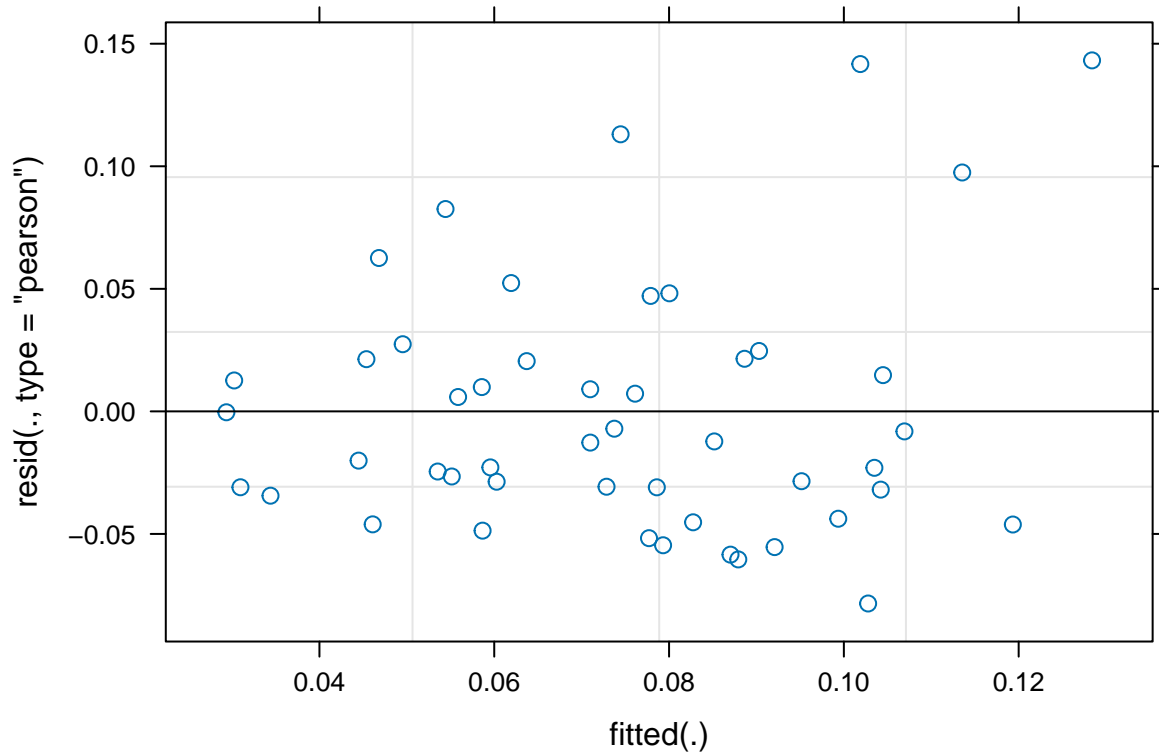
## Term: (Intercept)
##   T-value: 2.578255
##   Number of Observations: 48
##   P-value: 0.01312372
##
## Term: Condition
##   T-value: -0.6390206
##   Number of Observations: 48
##   P-value: 0.5259117
##
## Term: time
##   T-value: -1.522033
##   Number of Observations: 48
##   P-value: 0.134701
##
## Term: Condition:time
##   T-value: 0.7038559
##   Number of Observations: 48
##   P-value: 0.4849952

```

```

plot(model)

```



```
# Shapiro-Wilk test for normality
shapiro.test(residuals(model))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(model)
## W = 0.9039, p-value = 0.0008425
```

2.2.4 Communication with Indeterminate

```
library(readr)
library(lme4)

CircleTimeData <- read_csv("~/GitHub/CircleTime_HRI_DataAnalysis/CircleTimeData.csv")
```

```
## Rows: 48 Columns: 15
## -- Column specification -----
## Delimiter: ","
## dbl (15): Subject, Condition, time, Affect_Positive, Affect_Negative, Affect...
##
## 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.
```

```

# View(CircleTimeData)
# head(CircleTimeData)

# Fit a mixed-effects model

df <- CircleTimeData
# spec(df)
model <- lmer(Communication_with_Indeterminent ~ Condition * time + (1 | Subject), data = df)

```

```
## boundary (singular) fit: see help('isSingular')
```

```
summary(model)
```

```

## Linear mixed model fit by REML ['lmerMod']
## Formula: Communication_with_Indeterminent ~ Condition * time + (1 | Subject)
## Data: df
##
## REML criterion at convergence: -138.2
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.3838 -0.5372 -0.1631  0.2424  3.8207
##
## Random effects:
## Groups Name Variance Std.Dev.
## Subject (Intercept) 0.000000 0.00000
## Residual 0.001877 0.04332
## Number of obs: 48, groups: Subject, 6
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)  0.057975  0.048433  1.197
## Condition    0.011929  0.030632  0.389
## time        -0.022226  0.017685 -1.257
## Condition:time 0.004757  0.011185  0.425
##
## Correlation of Fixed Effects:
##              (Intr) Condtn time
## Condition    -0.949
## time         -0.913  0.866
## Condition:tm  0.866 -0.913 -0.949
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')

```

```

summary_table <- summary(model)$coefficients
t_values <- summary_table[, "t value"]

```

```

# Number of observations
num_obs <- nobs(model)

```

```
for (i in seq_along(t_values)) {
```

```

# Extract t-value for the term
t_value <- t_values[i]
num_obs_term <- num_obs

# Calculate p-value using t-value and degrees of freedom
df <- num_obs_term - 1 # degrees of freedom for a fixed effect
p_value <- 2 * (1 - pt(abs(t_value), df))

# Print or use the results as needed
cat("Term:", names(t_values)[i], "\n")
cat("  T-value:", t_value, "\n")
cat("  Number of Observations:", num_obs_term, "\n")
cat("  P-value:", p_value, "\n\n")
}

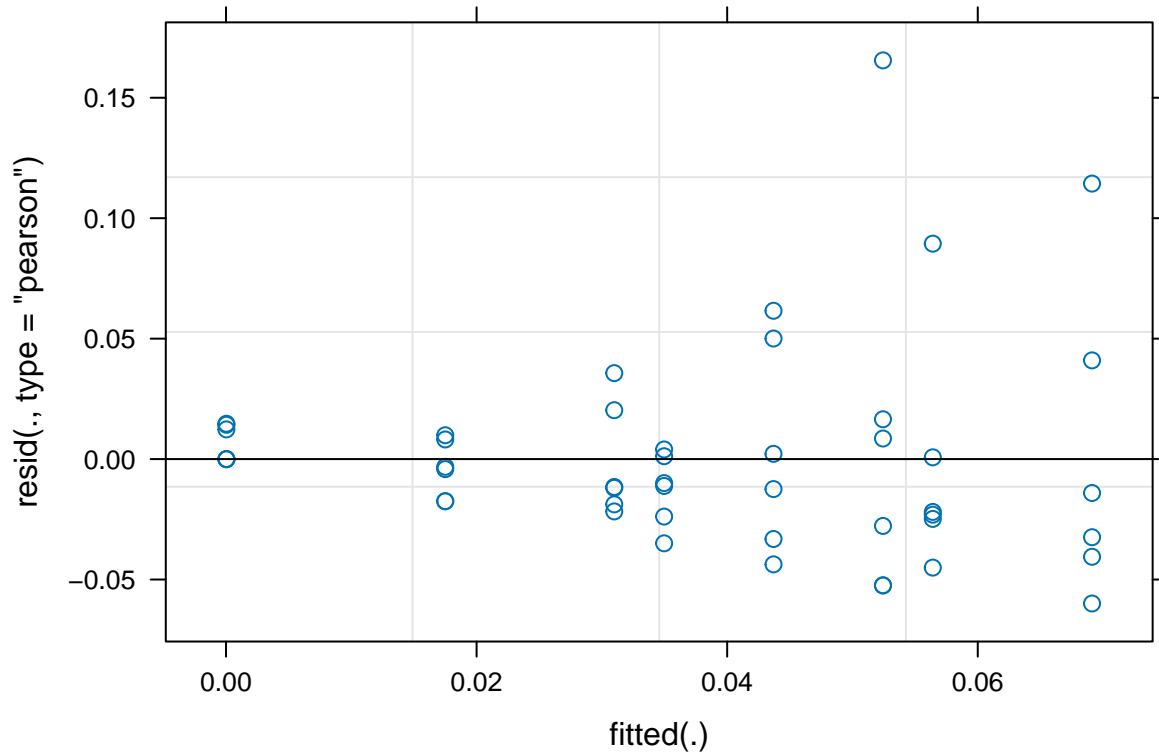
```

```

## Term: (Intercept)
##   T-value: 1.197022
##   Number of Observations: 48
##   P-value: 0.2373008
##
## Term: Condition
##   T-value: 0.3894367
##   Number of Observations: 48
##   P-value: 0.6987121
##
## Term: time
##   T-value: -1.256732
##   Number of Observations: 48
##   P-value: 0.2150609
##
## Term: Condition:time
##   T-value: 0.4253146
##   Number of Observations: 48
##   P-value: 0.6725485

```

```
plot(model)
```



```
# Shapiro-Wilk test for normality
shapiro.test(residuals(model))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(model)
## W = 0.84453, p-value = 1.562e-05
```

2.3 Engagement

2.3.1 Engagement with On Target

```
library(readr)
library(lme4)

CircleTimeData <- read_csv("~/GitHub/CircleTime_HRI_DataAnalysis/CircleTimeData.csv")
```

```
## Rows: 48 Columns: 15
## -- Column specification -----
## Delimiter: ","
## dbl (15): Subject, Condition, time, Affect_Positive, Affect_Negative, Affect...
##
```



```
## 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.
```

```
# View(CircleTimeData)
# head(CircleTimeData)

# Fit a mixed-effects model

df <- CircleTimeData
# spec(df)
model <- lmer(Engagement_OnTarget ~ Condition * time + (1 | Subject), data = df)
```

```
## boundary (singular) fit: see help('isSingular')
```

```
summary(model)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: Engagement_OnTarget ~ Condition * time + (1 | Subject)
## Data: df
##
## REML criterion at convergence: -85.6
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -3.6223 -0.6152  0.1731  0.4749  2.0034
##
## Random effects:
## Groups Name Variance Std.Dev.
## Subject (Intercept) 0.000000 0.00000
## Residual 0.006199 0.07873
## Number of obs: 48, groups: Subject, 6
##
## Fixed effects:
## Estimate Std. Error t value
## (Intercept) 0.67094 0.08803 7.622
## Condition 0.05442 0.05567 0.978
## time 0.04266 0.03214 1.327
## Condition:time -0.02029 0.02033 -0.998
##
## Correlation of Fixed Effects:
## (Intr) Condtm time
## Condition -0.949
## time -0.913 0.866
## Conditin:tm 0.866 -0.913 -0.949
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

```
summary_table <- summary(model)$coefficients
t_values <- summary_table[, "t value"]

# Number of observations
num_obs <- nobs(model)
```

```

for (i in seq_along(t_values)) {
  # Extract t-value for the term
  t_value <- t_values[i]
  num_obs_term <- num_obs

  # Calculate p-value using t-value and degrees of freedom
  df <- num_obs_term - 1 # degrees of freedom for a fixed effect
  p_value <- 2 * (1 - pt(abs(t_value), df))

  # Print or use the results as needed
  cat("Term:", names(t_values)[i], "\n")
  cat("  T-value:", t_value, "\n")
  cat("  Number of Observations:", num_obs_term, "\n")
  cat("  P-value:", p_value, "\n\n")
}

```

```

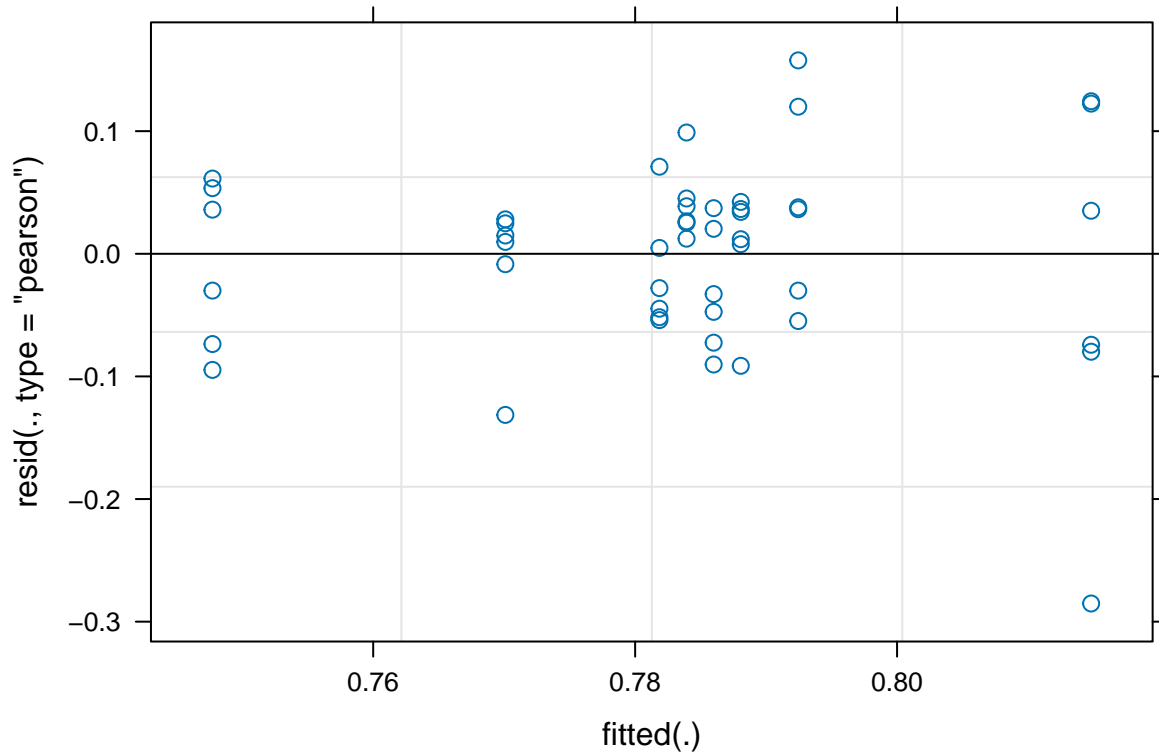
## Term: (Intercept)
##   T-value: 7.622177
##   Number of Observations: 48
##   P-value: 9.381269e-10
##
## Term: Condition
##   T-value: 0.9775059
##   Number of Observations: 48
##   P-value: 0.3333235
##
## Term: time
##   T-value: 1.327106
##   Number of Observations: 48
##   P-value: 0.1908848
##
## Term: Condition:time
##   T-value: -0.9982351
##   Number of Observations: 48
##   P-value: 0.3232772

```

```

plot(model)

```



```
# Shapiro-Wilk test for normality
shapiro.test(residuals(model))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(model)
## W = 0.93214, p-value = 0.00818
```

2.3.2 Engagement with Therapist

```
library(readr)
library(lme4)

CircleTimeData <- read_csv("~/GitHub/CircleTime_HRI_DataAnalysis/CircleTimeData.csv")
```

```
## Rows: 48 Columns: 15
## -- Column specification -----
## Delimiter: ","
## dbl (15): Subject, Condition, time, Affect_Positive, Affect_Negative, Affect...
##
## 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.
```

```

# View(CircleTimeData)
# head(CircleTimeData)

# Fit a mixed-effects model

df <- CircleTimeData
# spec(df)
model <- lmer(Engagement_Therapist ~ Condition * time + (1 | Subject), data = df)
summary(model)

```

```

## Linear mixed model fit by REML ['lmerMod']
## Formula: Engagement_Therapist ~ Condition * time + (1 | Subject)
## Data: df
##
## REML criterion at convergence: -177.7
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.49802 -0.75223 -0.00126  0.61619  2.50162
##
## Random effects:
## Groups Name Variance Std.Dev.
## Subject (Intercept) 1.373e-06 0.001172
## Residual 7.643e-04 0.027647
## Number of obs: 48, groups: Subject, 6
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)  0.049389  0.030914  1.598
## Condition    0.007233  0.019549  0.370
## time         0.003111  0.011287  0.276
## Condition:time -0.003408  0.007138 -0.477
##
## Correlation of Fixed Effects:
##              (Intr) Condtn time
## Condition   -0.949
## time        -0.913  0.866
## Conditin:tm  0.866 -0.913 -0.949

```

```

summary_table <- summary(model)$coefficients
t_values <- summary_table[, "t value"]

# Number of observations
num_obs <- nobs(model)

for (i in seq_along(t_values)) {
  # Extract t-value for the term
  t_value <- t_values[i]
  num_obs_term <- num_obs

  # Calculate p-value using t-value and degrees of freedom
  df <- num_obs_term - 1 # degrees of freedom for a fixed effect
}

```

```

p_value <- 2 * (1 - pt(abs(t_value), df))

# Print or use the results as needed
cat("Term:", names(t_values)[i], "\n")
cat("  T-value:", t_value, "\n")
cat("  Number of Observations:", num_obs_term, "\n")
cat("  P-value:", p_value, "\n\n")
}

```

```

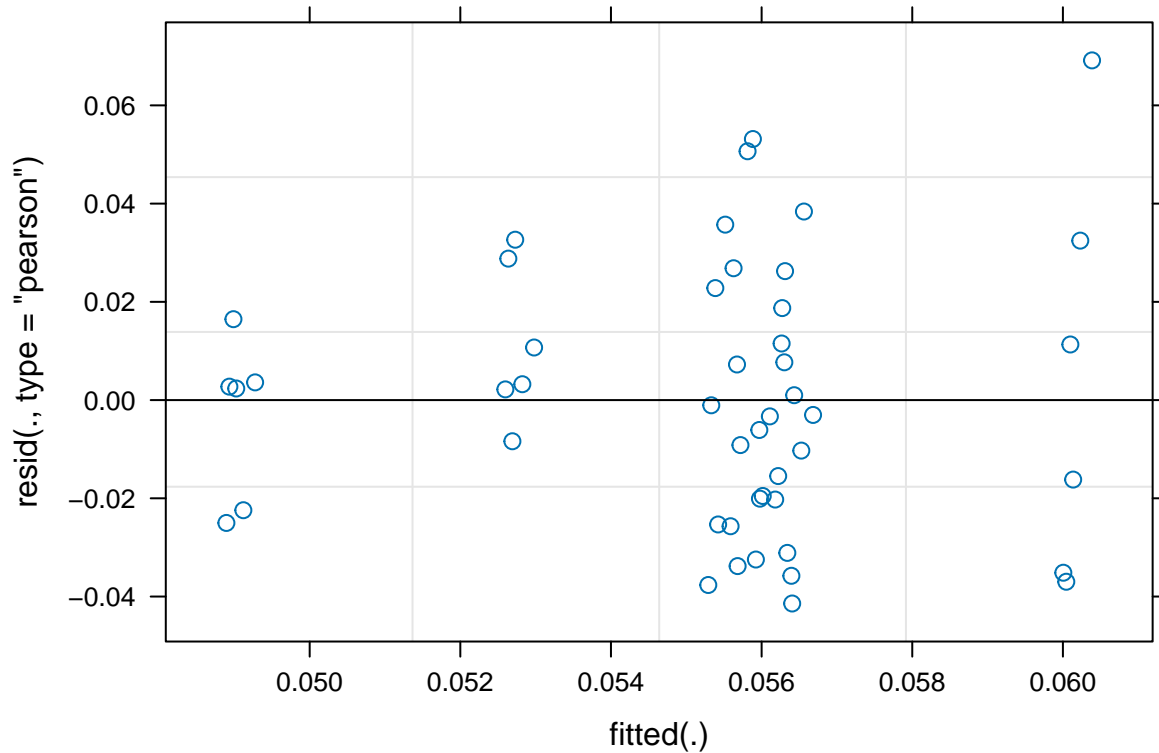
## Term: (Intercept)
##   T-value: 1.597641
##   Number of Observations: 48
##   P-value: 0.1168253
##
## Term: Condition
##   T-value: 0.3699688
##   Number of Observations: 48
##   P-value: 0.7130677
##
## Term: time
##   T-value: 0.2756717
##   Number of Observations: 48
##   P-value: 0.7840087
##
## Term: Condition:time
##   T-value: -0.4773599
##   Number of Observations: 48
##   P-value: 0.6353183

```

```

plot(model)

```



```
# Shapiro-Wilk test for normality
shapiro.test(residuals(model))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(model)
## W = 0.96631, p-value = 0.1811
```

2.3.3 Engagement with OffTarget

```
library(readr)
library(lme4)

CircleTimeData <- read_csv("~/GitHub/CircleTime_HRI_DataAnalysis/CircleTimeData.csv")
```

```
## Rows: 48 Columns: 15
## -- Column specification -----
## Delimiter: ","
## dbl (15): Subject, Condition, time, Affect_Positive, Affect_Negative, Affect...
##
## 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.
```

```

# View(CircleTimeData)
# head(CircleTimeData)

# Fit a mixed-effects model

df <- CircleTimeData
# spec(df)
model <- lmer(Engagement_OffTarget ~ Condition * time + (1 | Subject), data = df)
summary(model)

```

```

## Linear mixed model fit by REML ['lmerMod']
## Formula: Engagement_OffTarget ~ Condition * time + (1 | Subject)
## Data: df
##
## REML criterion at convergence: -98.4
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.9573 -0.4189 -0.1439  0.4633  4.1382
##
## Random effects:
## Groups Name Variance Std.Dev.
## Subject (Intercept) 9.676e-05 0.009837
## Residual 4.559e-03 0.067524
## Number of obs: 48, groups: Subject, 6
##
## Fixed effects:
## Estimate Std. Error t value
## (Intercept) 0.28119 0.07560 3.719
## Condition -0.06322 0.04775 -1.324
## time -0.04668 0.02757 -1.693
## Condition:time 0.02464 0.01743 1.413
##
## Correlation of Fixed Effects:
## (Intr) Condtm time
## Condition -0.947
## time -0.912 0.866
## Conditin:tm 0.865 -0.913 -0.949

```

```

summary_table <- summary(model)$coefficients
t_values <- summary_table[, "t value"]

# Number of observations
num_obs <- nobs(model)

for (i in seq_along(t_values)) {
  # Extract t-value for the term
  t_value <- t_values[i]
  num_obs_term <- num_obs

  # Calculate p-value using t-value and degrees of freedom
  df <- num_obs_term - 1 # degrees of freedom for a fixed effect
}

```

```

p_value <- 2 * (1 - pt(abs(t_value), df))

# Print or use the results as needed
cat("Term:", names(t_values)[i], "\n")
cat("  T-value:", t_value, "\n")
cat("  Number of Observations:", num_obs_term, "\n")
cat("  P-value:", p_value, "\n\n")
}

```

```

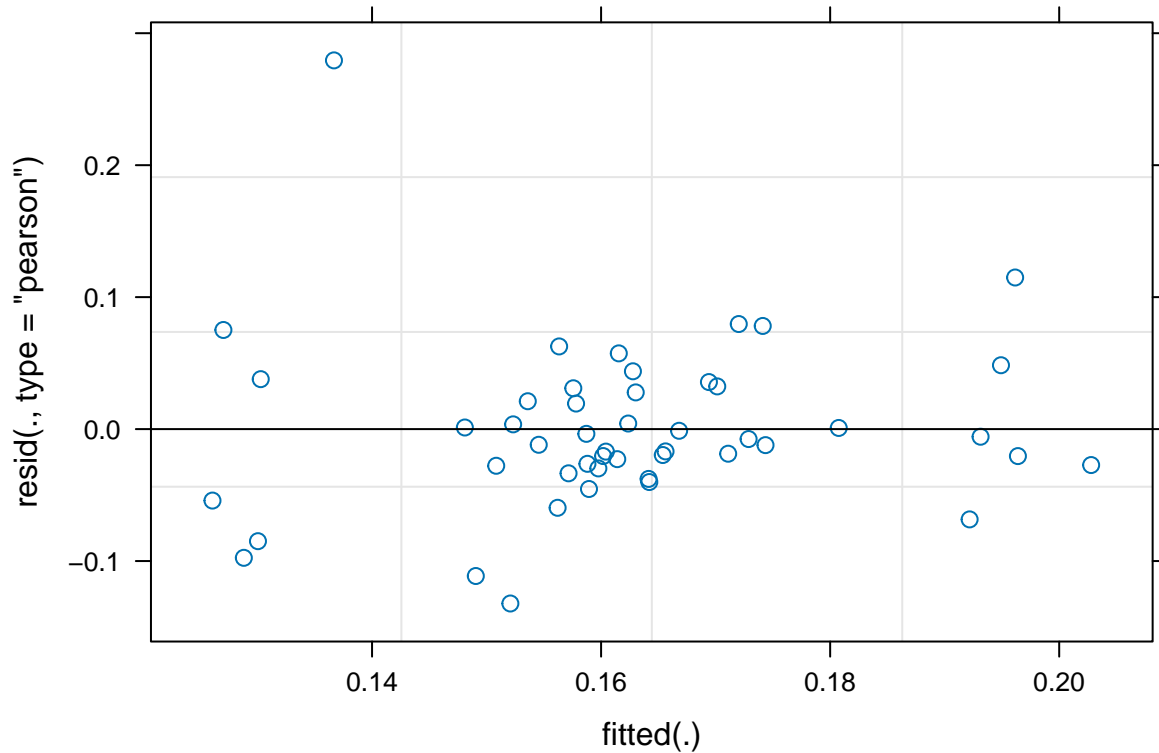
## Term: (Intercept)
##   T-value: 3.719353
##   Number of Observations: 48
##   P-value: 0.0005319465
##
## Term: Condition
##   T-value: -1.324088
##   Number of Observations: 48
##   P-value: 0.1918773
##
## Term: time
##   T-value: -1.693314
##   Number of Observations: 48
##   P-value: 0.0970147
##
## Term: Condition:time
##   T-value: 1.413353
##   Number of Observations: 48
##   P-value: 0.164141

```

```

plot(model)

```

```
# Shapiro-Wilk test for normality
shapiro.test(residuals(model))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(model)
## W = 0.88565, p-value = 0.0002234
```

2.4 Performance

2.4.1 Performance Positive

```
library(readr)
library(lme4)

CircleTimeData <- read_csv("~/GitHub/CircleTime_HRI_DataAnalysis/CircleTimeData.csv")

## Rows: 48 Columns: 15
## -- Column specification -----
## Delimiter: ","
## dbl (15): Subject, Condition, time, Affect_Positive, Affect_Negative, Affect...
##
```

```
## 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.
```

```
# View(CircleTimeData)
# head(CircleTimeData)

# Fit a mixed-effects model

df <- CircleTimeData
# spec(df)
model <- lmer(Performance_Positive ~ Condition * time + (1 | Subject), data = df)
summary(model)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: Performance_Positive ~ Condition * time + (1 | Subject)
## Data: df
##
## REML criterion at convergence: -50.4
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.03478 -0.65951  0.04353  0.55740  1.72469
##
## Random effects:
## Groups Name Variance Std.Dev.
## Subject (Intercept) 0.030709 0.17524
## Residual 0.009498 0.09746
## Number of obs: 48, groups: Subject, 6
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)  0.876364  0.130349  6.723
## Condition   -0.082580  0.068914 -1.198
## time         0.002000  0.039787  0.050
## Condition:time 0.005816  0.025164  0.231
##
## Correlation of Fixed Effects:
##              (Intr) Condtn time
## Condition   -0.793
## time        -0.763  0.866
## Conditin:tm  0.724 -0.913 -0.949
```

```
summary_table <- summary(model)$coefficients
t_values <- summary_table[, "t value"]
```

```
# Number of observations
num_obs <- nobs(model)
```

```
for (i in seq_along(t_values)) {
  # Extract t-value for the term
  t_value <- t_values[i]
  num_obs_term <- num_obs
```

```

# Calculate p-value using t-value and degrees of freedom
df <- num_obs_term - 1 # degrees of freedom for a fixed effect
p_value <- 2 * (1 - pt(abs(t_value), df))

# Print or use the results as needed
cat("Term:", names(t_values)[i], "\n")
cat("  T-value:", t_value, "\n")
cat("  Number of Observations:", num_obs_term, "\n")
cat("  P-value:", p_value, "\n\n")
}

```

```

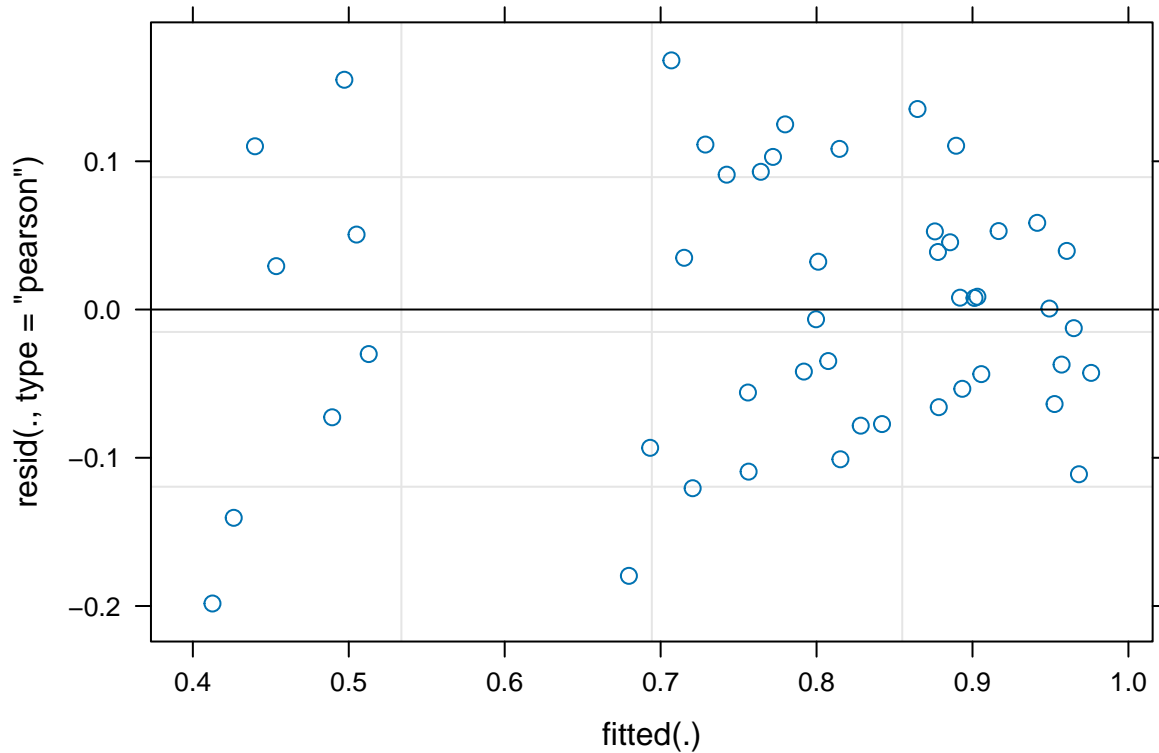
## Term: (Intercept)
##   T-value: 6.723202
##   Number of Observations: 48
##   P-value: 2.150543e-08
##
## Term: Condition
##   T-value: -1.198313
##   Number of Observations: 48
##   P-value: 0.2368031
##
## Term: time
##   T-value: 0.05026613
##   Number of Observations: 48
##   P-value: 0.9601233
##
## Term: Condition:time
##   T-value: 0.231129
##   Number of Observations: 48
##   P-value: 0.8182179

```

```

plot(model)

```



```
# Shapiro-Wilk test for normality
shapiro.test(residuals(model))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(model)
## W = 0.98221, p-value = 0.673
```

2.4.2 Performance Negative

```
library(readr)
library(lme4)

CircleTimeData <- read_csv("~/GitHub/CircleTime_HRI_DataAnalysis/CircleTimeData.csv")
```

```
## Rows: 48 Columns: 15
## -- Column specification -----
## Delimiter: ","
## dbl (15): Subject, Condition, time, Affect_Positive, Affect_Negative, Affect...
##
## 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.
```

```

# View(CircleTimeData)
# head(CircleTimeData)

# Fit a mixed-effects model

df <- CircleTimeData
# spec(df)
model <- lmer(Performance_Negative ~ Condition * time + (1 | Subject), data = df)
summary(model)

```

```

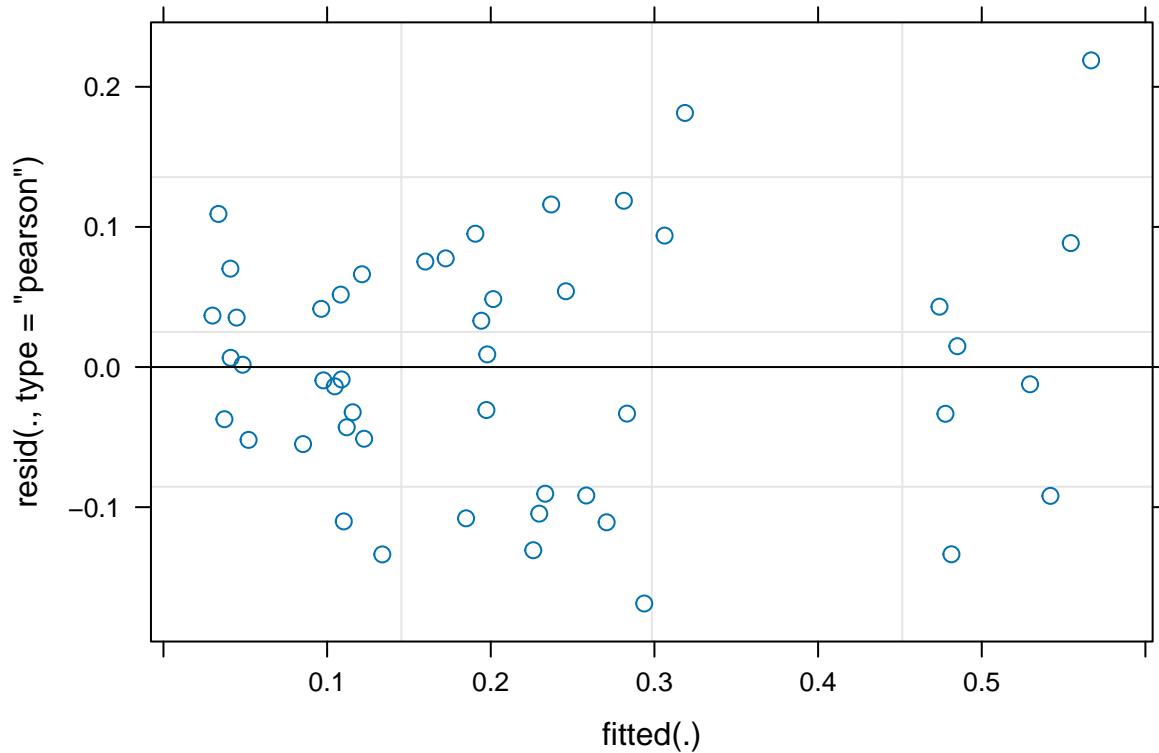
## Linear mixed model fit by REML ['lmerMod']
## Formula: Performance_Negative ~ Condition * time + (1 | Subject)
## Data: df
##
## REML criterion at convergence: -53
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.78028 -0.55648 -0.03801  0.60251  2.30889
##
## Random effects:
## Groups Name Variance Std.Dev.
## Subject (Intercept) 0.028326 0.1683
## Residual 0.008986 0.0948
## Number of obs: 48, groups: Subject, 6
##
## Fixed effects:
## Estimate Std. Error t value
## (Intercept) 0.101811 0.126308 0.806
## Condition 0.090517 0.067031 1.350
## time 0.005143 0.038700 0.133
## Condition:time -0.008792 0.024476 -0.359
##
## Correlation of Fixed Effects:
## (Intr) Condtn time
## Condition -0.796
## time -0.766 0.866
## Conditin:tm 0.727 -0.913 -0.949

```

```

plot(model)

```



```
summary_table <- summary(model)$coefficients
t_values <- summary_table[, "t value"]

# Number of observations
num_obs <- nobs(model)

for (i in seq_along(t_values)) {
  # Extract t-value for the term
  t_value <- t_values[i]
  num_obs_term <- num_obs

  # Calculate p-value using t-value and degrees of freedom
  df <- num_obs_term - 1 # degrees of freedom for a fixed effect
  p_value <- 2 * (1 - pt(abs(t_value), df))

  # Print or use the results as needed
  cat("Term:", names(t_values)[i], "\n")
  cat("  T-value:", t_value, "\n")
  cat("  Number of Observations:", num_obs_term, "\n")
  cat("  P-value:", p_value, "\n\n")
}
```

```
## Term: (Intercept)
##   T-value: 0.8060472
##   Number of Observations: 48
```

```
## P-value: 0.4242753
##
## Term: Condition
## T-value: 1.350373
## Number of Observations: 48
## P-value: 0.1833632
##
## Term: time
## T-value: 0.1328902
## Number of Observations: 48
## P-value: 0.8948475
##
## Term: Condition:time
## T-value: -0.3592163
## Number of Observations: 48
## P-value: 0.7210422
```

```
# Shapiro-Wilk test for normality
shapiro.test(residuals(model))
```

```
##
## Shapiro-Wilk normality test
##
## data: residuals(model)
## W = 0.98303, p-value = 0.7081
```