

PP_Narrative_Analysis_H10

Kristen Johnson

2025-04-04

```
# install necessary packages
```

```
#install.packages("lmerTest")
```

```
# Load required packages
```

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
```

```
## v dplyr      1.1.4      v readr      2.1.5
```

```
## v forcats    1.0.0      v stringr    1.5.1
```

```
## v ggplot2    3.5.1      v tibble     3.2.1
```

```
## v lubridate  1.9.4      v tidyr      1.3.1
```

```
## v purrr      1.0.4
```

```
## -- Conflicts ----- tidyverse_conflicts() --
```

```
## x dplyr::filter() masks stats::filter()
```

```
## x dplyr::lag()     masks stats::lag()
```

```
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(car)      # For regression diagnostics
```

```
## Loading required package: carData
```

```
##
```

```
## Attaching package: 'car'
```

```
##
```

```
## The following object is masked from 'package:dplyr':
```

```
##
```

```
##      recode
```

```
##
```

```
## The following object is masked from 'package:purrr':
```

```
##
```

```
##      some
```

```
library(lmtest)    # For regression diagnostics
```

```
## Loading required package: zoo
```

```
##
```

```
## Attaching package: 'zoo'
```

```
##
```

```
## The following objects are masked from 'package:base':
```

```
##
```

```
##      as.Date, as.Date.numeric
```

```
library(effectsize) # For effect sizes
```

```
library(ggplot2)    # For visualization
```

```
library(here)
```

```

## here() starts at /Users/kristenjohnson/KristenWorkingDirectory/Play_Narrative
library(dplyr)
library(lme4)

## Loading required package: Matrix
##
## Attaching package: 'Matrix'
##
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
library(lmerTest)

##
## Attaching package: 'lmerTest'
##
## The following object is masked from 'package:lme4':
##
##     lmer
##
## The following object is masked from 'package:stats':
##
##     step
library(extrafont)

## Registering fonts with R
# find project root directory automatically
merged_H10 <- read.csv(here("PN_Datasets", "Pretend_Play", "CSVs_of_Combined_Data_PP", "merged_H10.csv"))

H10_gesture_play <- merged_H10 %>%
  group_by(participant_id) %>%
  summarise(
    total_pretend_play = sum(c_pret, na.rm = TRUE),
    mlu = mean(C_wpu, na.rm = TRUE),
    any_pretend_play = if_else(any(c_pret == 1, na.rm = TRUE), 1, 0),
    total_rep_gestures = sum(gesture_rep == 1, na.rm = TRUE),
    total_all_gestures = sum(gesture_all == 1, na.rm = TRUE),
    group_status = first(GroupStatus)
  )

write_csv(H10_gesture_play,
  here("PN_DataAnalysis", "PP_Narrative_Analysis", "H10_gesture_play.csv"))

# read in data frame that is just gesture during instance of pretend play
pretend_data_H10 <- read.csv(here("PN_Datasets", "Pretend_Play", "CSVs_of_Combined_Data_PP", "pretend_data_H10.csv"))

# Aggregate by child
child_summary_H10 <- pretend_data_H10 %>%
  group_by(participant_id, GroupStatus) %>%
  summarize(
    total_gestures = sum(gesture_all),

```

```

rep_gestures = sum(gesture_rep),
icon_gestures = sum(gesture_icon),
prop_rep = sum(gesture_rep) / sum(gesture_all),
prop_icon = sum(gesture_icon) / sum(gesture_all),
)

## `summarise()` has grouped output by 'participant_id'. You can override using
## the `.groups` argument.

# add mlu from all utterances

pretend_data_H10 <- child_summary_H10 %>%
  left_join(select(H10_gesture_play, participant_id, mlu), by = "participant_id") %>%
  left_join(select(H10_gesture_play, participant_id, total_pretend_play), by = "participant_id")

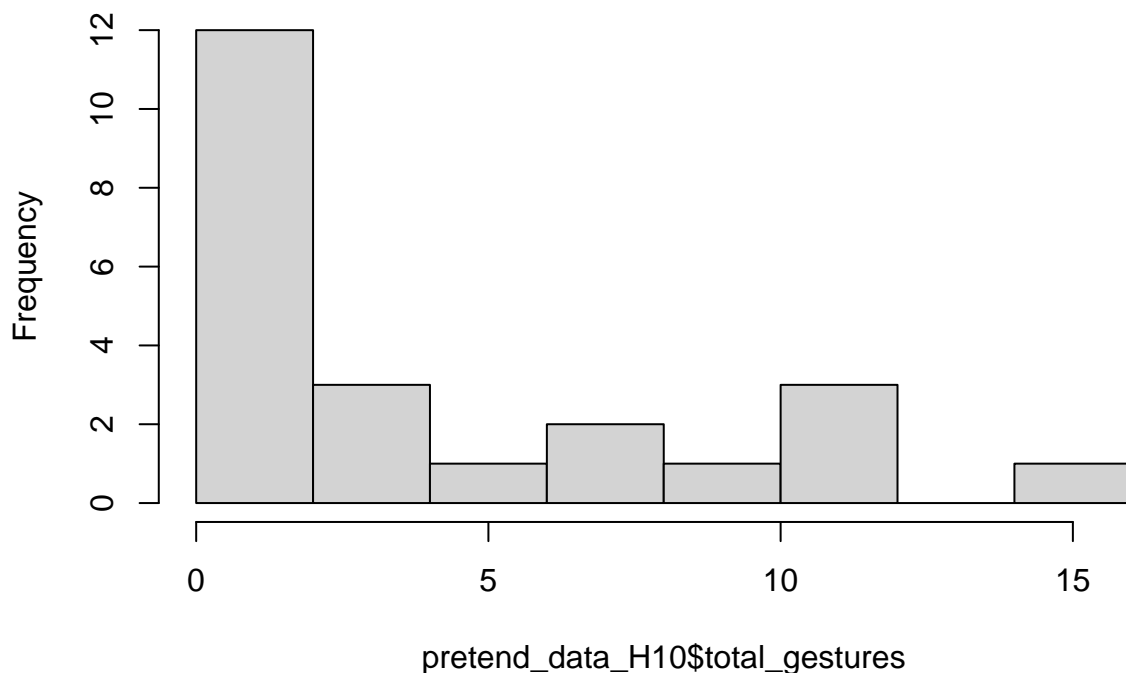
write_csv(pretend_data_H10, here("PN_DataAnalysis", "PP_Narrative_Analysis", "pretend_data_H10.csv"))

# Create binary variable for representational gesture presence
pretend_data_H10 <- pretend_data_H10 %>%
  mutate(rep_gesture_present = ifelse(rep_gestures > 0, 1, 0))

# Check distributions
hist(pretend_data_H10$total_gestures)

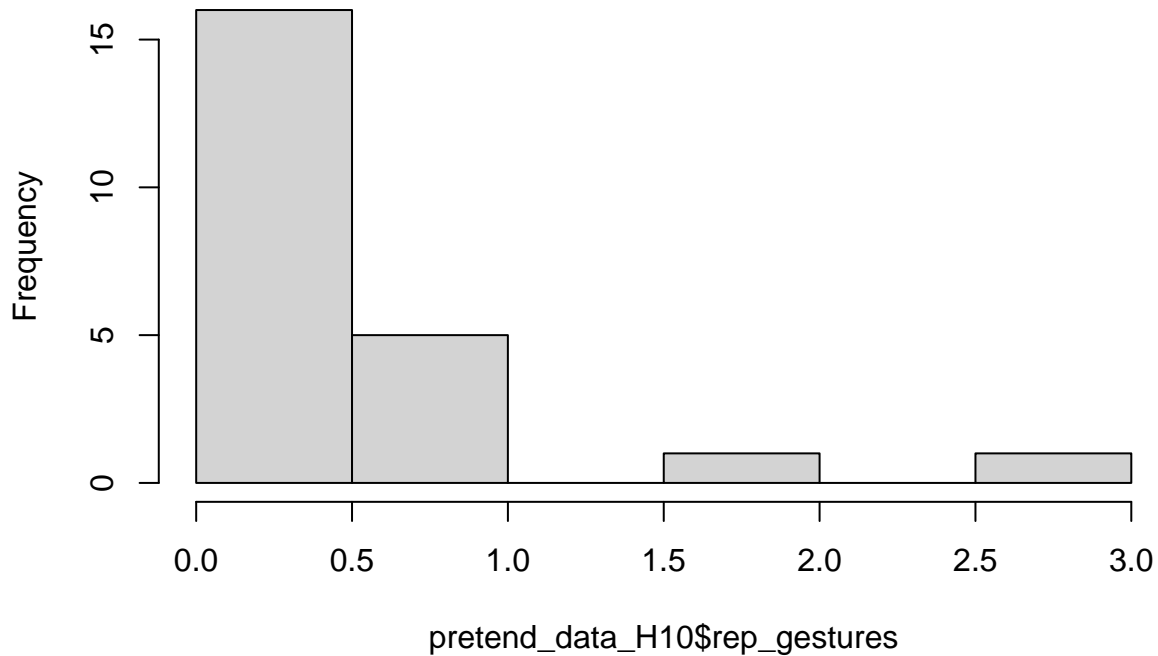
```

Histogram of pretend_data_H10\$total_gestures



```
hist(pretend_data_H10$rep_gestures)
```

Histogram of pretend_data_H10\$rep_gestures



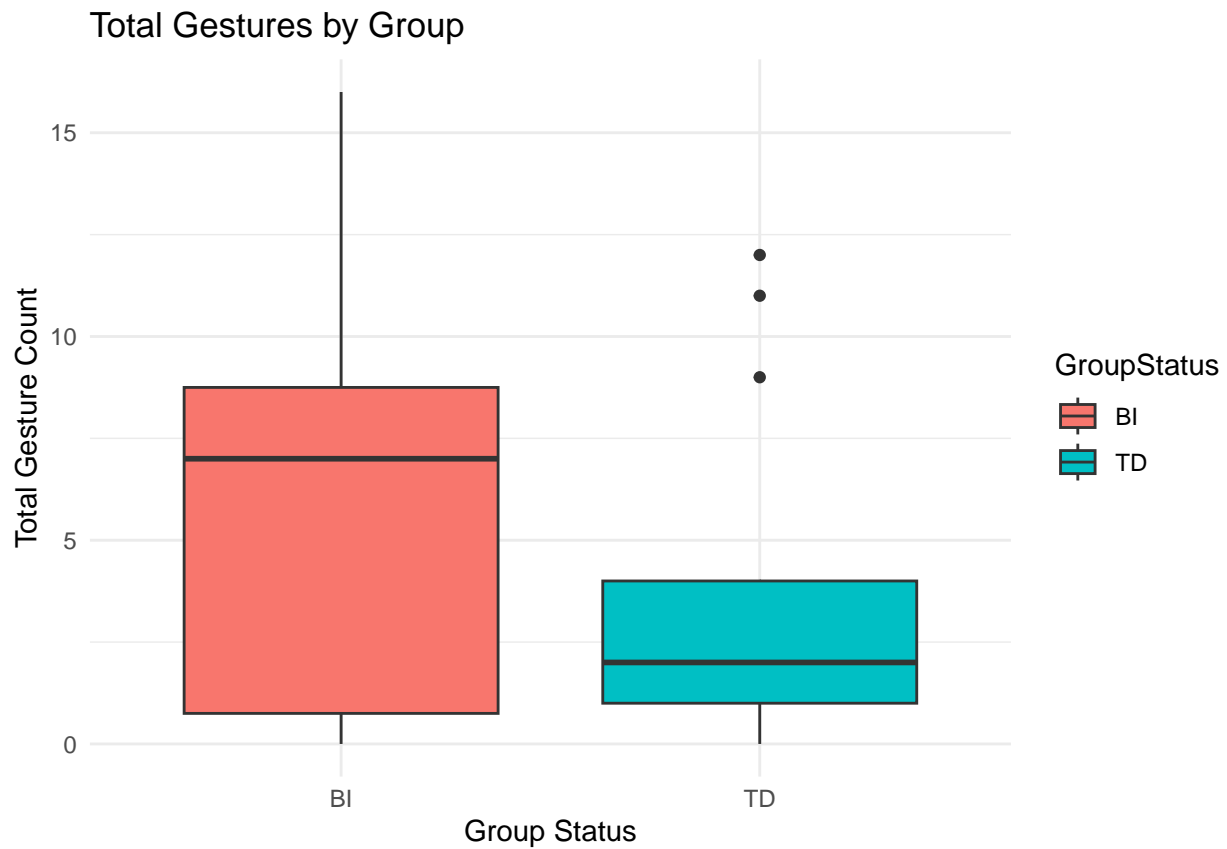
```
table(pretend_data_H10$rep_gesture_present, pretend_data_H10$GroupStatus)
```

```
##
##      BI TD
##    0  7  9
##    1  1  6
```

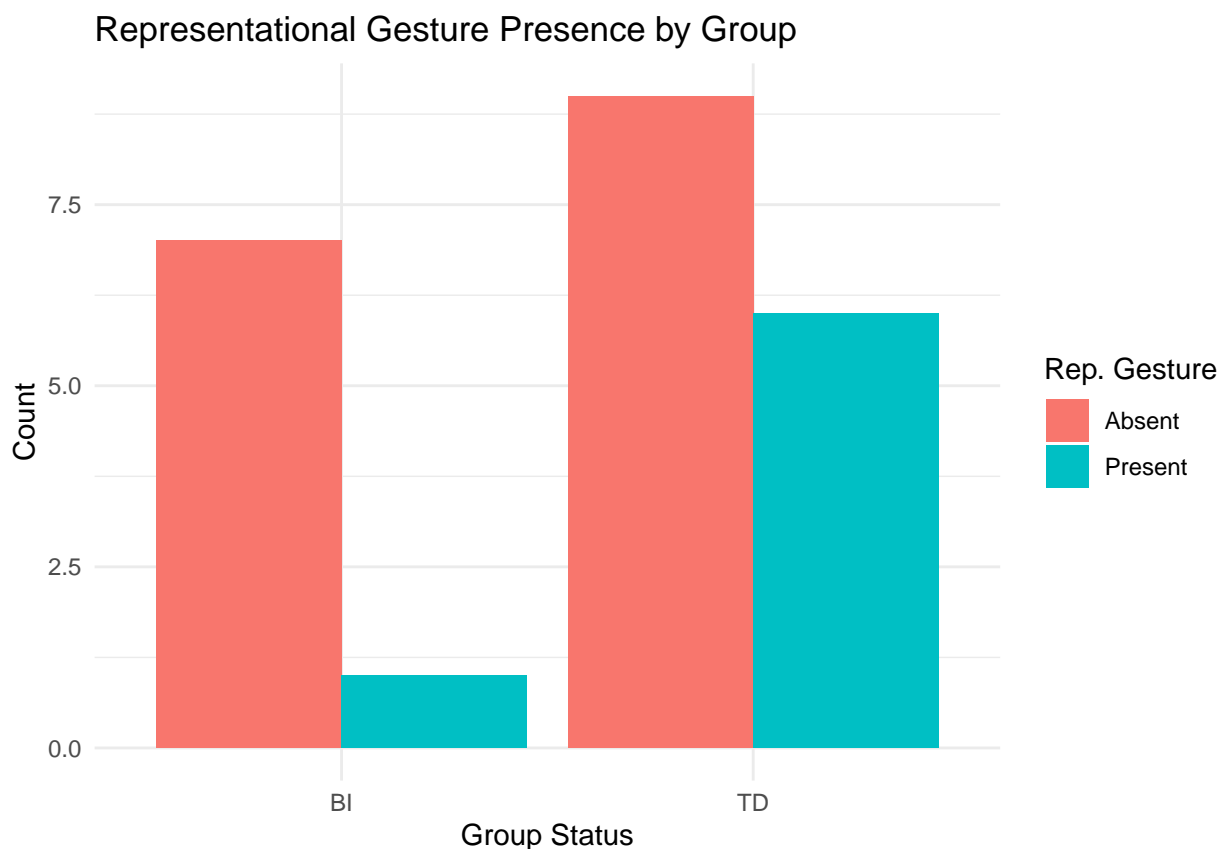
```
# Check correlation between total gestures and representational gestures
cor.test(pretend_data_H10$total_gestures, pretend_data_H10$rep_gestures)
```

```
##
## Pearson's product-moment correlation
##
## data:  pretend_data_H10$total_gestures and pretend_data_H10$rep_gestures
## t = 1.6213, df = 21, p-value = 0.1199
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.09119772  0.65560754
## sample estimates:
##      cor
## 0.333543
```

```
# 2. Data visualization
# Boxplot of total gestures by group
ggplot(pretend_data_H10, aes(x = GroupStatus, y = total_gestures, fill = GroupStatus)) +
  geom_boxplot() +
  theme_minimal() +
  labs(title = "Total Gestures by Group",
       x = "Group Status",
       y = "Total Gesture Count")
```



```
# Bar chart of representational gesture presence by group
ggplot(pretend_data_H10, aes(x = GroupStatus, fill = factor(rep_gesture_present))) +
  geom_bar(position = "dodge") +
  scale_fill_discrete(name = "Rep. Gesture", labels = c("Absent", "Present")) +
  theme_minimal() +
  labs(title = "Representational Gesture Presence by Group",
       x = "Group Status",
       y = "Count")
```



3. Merge with narrative outcome data

```
narrative_data <- read.csv(here("PN_Datasets", "Narrative", "CSVs_of_Combined_Data_N", "Narrative_Resul

# Rename participant column
narrative_data <- narrative_data %>%
  rename(participant_id = ChildID)

write.csv(narrative_data, here("PN_DataAnalysis", "PP_Narrative_Analysis", "narrative_data.csv"))

# MERGE IT
completely_merged_data_H10 <- left_join(pretend_data_H10, narrative_data, by = "participant_id") %>%
  mutate(GroupStatus = if_else(GroupStatus == "BI", "PL", GroupStatus))

write_csv(completely_merged_data_H10, here("PN_DataAnalysis", "PP_Narrative_Analysis", "completely_merg
```

4. Run regression models

```
# Model 1: Total gestures as predictor
model1 <- lm(AvgHighScore ~ GroupStatus + total_gestures + total_pretend_play + mlu + GroupStatus:total.
  data = completely_merged_data_H10)

# Model 2: Representational gesture presence as predictor
model2 <- lm(AvgHighScore ~ GroupStatus + rep_gesture_present + total_pretend_play + mlu + GroupStatus:
  data = completely_merged_data_H10)

summary(model1)
```

```
##
## Call:
## lm(formula = AvgHighScore ~ GroupStatus + total_gestures + total_pretend_play +
##      mlu + GroupStatus:total_gestures, data = completely_merged_data_H10)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.2419 -0.9897 -0.2255  1.3564  3.5751
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.891324   2.406845   0.370   0.7160
## GroupStatusTD    3.147592   2.049994   1.535   0.1442
## total_gestures    0.181573   0.204402   0.888   0.3875
## total_pretend_play 0.007074   0.012266   0.577   0.5721
## mlu             1.291709   0.684936   1.886   0.0776 .
## GroupStatusTD:total_gestures -0.275815   0.248960  -1.108   0.2843
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.13 on 16 degrees of freedom
## (1 observation deleted due to missingness)
## Multiple R-squared:  0.2935, Adjusted R-squared:  0.07275
## F-statistic:  1.33 on 5 and 16 DF,  p-value: 0.3016
```

```
summary(model2)
```

```
##
## Call:
## lm(formula = AvgHighScore ~ GroupStatus + rep_gesture_present +
##      total_pretend_play + mlu + GroupStatus:rep_gesture_present,
##      data = completely_merged_data_H10)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.2993 -0.8215 -0.5212  1.2233  3.4835
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.571599   1.733813   0.906   0.3782
## GroupStatusTD    2.288758   1.364309   1.678   0.1128
## rep_gesture_present 4.805092   2.076354   2.314   0.0343 *
## total_pretend_play 0.001307   0.010419   0.125   0.9017
## mlu             1.332648   0.620081   2.149   0.0473 *
## GroupStatusTD:rep_gesture_present -5.350047   2.295726  -2.330   0.0332 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.898 on 16 degrees of freedom
## (1 observation deleted due to missingness)
## Multiple R-squared:  0.4389, Adjusted R-squared:  0.2636
## F-statistic:  2.503 on 5 and 16 DF,  p-value: 0.07404
```

```
# Calculate effect sizes
eta_squared(model1)
```

```
## # Effect Size for ANOVA (Type I)
##
## Parameter | Eta2 (partial) | 95% CI
## -----
## GroupStatus | 0.10 | [0.00, 1.00]
## total_gestures | 3.49e-03 | [0.00, 1.00]
## total_pretend_play | 6.07e-03 | [0.00, 1.00]
## mlu | 0.18 | [0.00, 1.00]
## GroupStatus:total_gestures | 0.07 | [0.00, 1.00]
##
## - One-sided CIs: upper bound fixed at [1.00].
```

```
eta_squared(model2)
```

```
## # Effect Size for ANOVA (Type I)
##
## Parameter | Eta2 (partial) | 95% CI
## -----
## GroupStatus | 0.13 | [0.00, 1.00]
## rep_gesture_present | 1.41e-06 | [0.00, 1.00]
## total_pretend_play | 5.53e-03 | [0.00, 1.00]
## mlu | 0.23 | [0.00, 1.00]
## GroupStatus:rep_gesture_present | 0.25 | [0.01, 1.00]
##
## - One-sided CIs: upper bound fixed at [1.00].
```

5b. USE WHEN HAVE ALL DATA FROM 3 TIMEPOINTS.... mixed effects models

```
#library(lme4)
#library(lmerTest)

#mixed_model1 <- lmer(AvgHighScore ~ GroupStatus + total_gestures + total_pretend_play + mlu +
# GroupStatus:total_gestures + (1/participant_id),
# data = completely_merged_data)

#mixed_model2 <- lmer(AvgHighScore ~ GroupStatus + rep_gesture_present + total_pretend_play + mlu +
#GroupStatus:rep_gesture_present + (1/participant_id),
#data = completely_merged_data)

#summary(mixed_model1)
#summary(mixed_model2)

# Interaction Plot: Group × Representational Gesture Presence
#| label: fig-interaction
#| fig-cap: "Predicted narrative scores by group and representational gesture use. Error bars represent
#| fig-width: 6
#| fig-height: 4
#|
options(warn = 2)
```



```

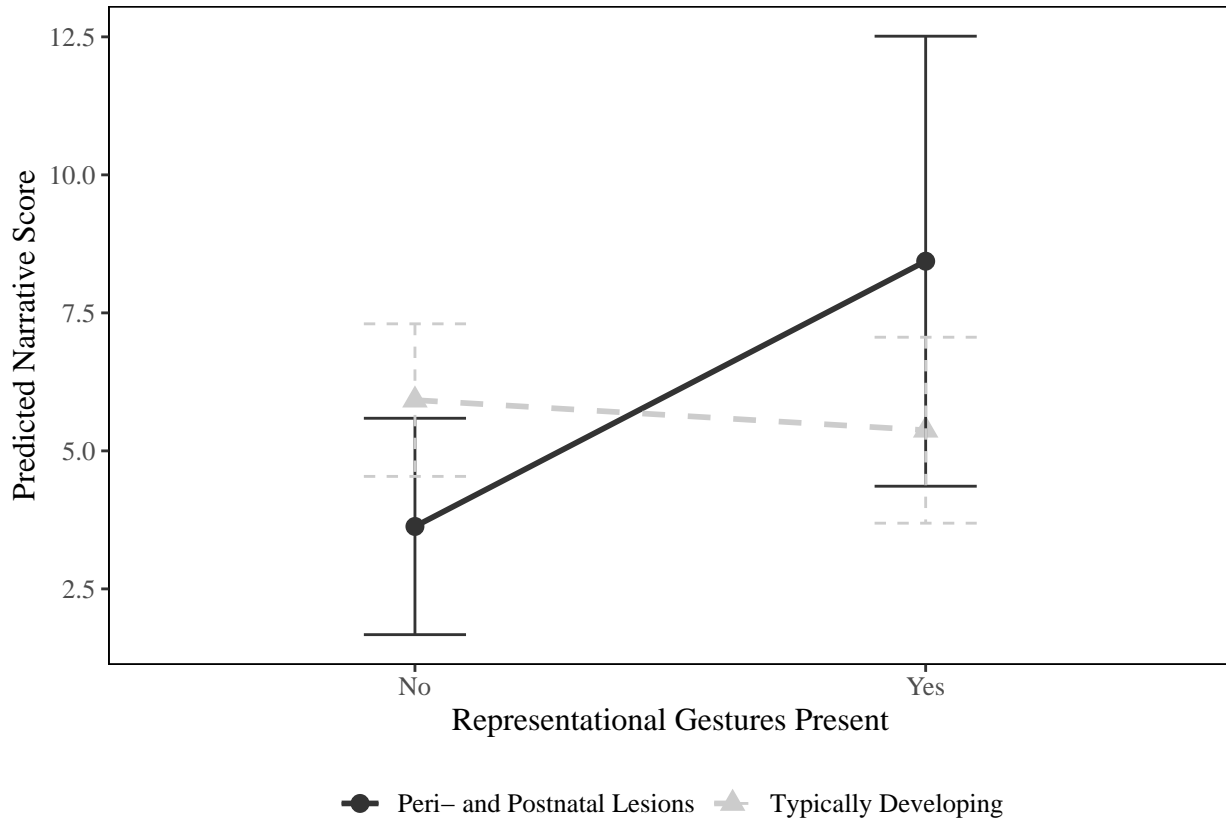
# Create a data frame for prediction (numeric variables for model compatibility)
newdata <- expand.grid(
  GroupStatus = c("PL", "TD"),
  rep_gesture_present = c(0, 1),
  total_pretend_play = mean(completely_merged_data_H10$total_pretend_play, na.rm = TRUE),
  mlu = mean(completely_merged_data_H10$mlu, na.rm = TRUE)
)

# Get predictions and standard errors from the model
newdata$predicted <- predict(model2, newdata = newdata)
pred_se <- predict(model2, newdata = newdata, se.fit = TRUE)$se.fit
newdata$lower <- newdata$predicted - 1.96 * pred_se
newdata$upper <- newdata$predicted + 1.96 * pred_se

# Relabel variables as factors for plotting clarity
newdata$GroupStatus <- factor(newdata$GroupStatus,
  levels = c("PL", "TD"),
  labels = c("Peri- and Postnatal Lesions", "Typically Developing"))
newdata$rep_gesture_present <- factor(newdata$rep_gesture_present,
  levels = c(0, 1),
  labels = c("No", "Yes"))

# Create the interaction plot
ggplot(newdata, aes(x = rep_gesture_present, y = predicted,
  group = GroupStatus, color = GroupStatus,
  linetype = GroupStatus, shape = GroupStatus)) +
  geom_line(linewidth = 1) +
  geom_point(size = 3) +
  geom_errorbar(aes(ymin = lower, ymax = upper), width = 0.2) +
  labs(x = "Representational Gestures Present",
    y = "Predicted Narrative Score") +
  scale_color_grey(start = 0.2, end = 0.8) +
  scale_linetype_manual(values = c("solid", "dashed")) +
  scale_shape_manual(values = c(16, 17)) +
  theme_bw() +
  theme(
    text = element_text(family = "Times New Roman", size = 12),
    axis.title = element_text(size = 12),
    axis.text = element_text(size = 10),
    legend.position = "bottom",
    legend.title = element_blank(),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    panel.border = element_rect(color = "black")
  )
)

```



Scatter Plot: Raw Data with Regression Lines

#| label: fig-scatter #| fig-cap: "Relationship between representational gesture presence and narrative scores by group. Shaded areas represent 95% confidence intervals around regression lines." #| fig-width: 7 #| fig-height: 4

```
ggplot(completely_merged_data_H10, aes(x = as.factor(rep_gesture_present), y = AvgHighScore, fill = GroupStatus, shape = GroupStatus)) + geom_jitter(width = 0.1, height = 0, size = 3, alpha = 0.7) + geom_smooth(aes(group = GroupStatus, linetype = GroupStatus, color = GroupStatus), method = "lm", se = TRUE, formula = y ~ x) + scale_x_discrete(labels = c("0" = "No", "1" = "Yes")) + scale_shape_manual(values = c(21, 24)) + scale_fill_grey(start = 0.5, end = 0.8) + scale_color_grey(start = 0.2, end = 0.6) + scale_linetype_manual(values = c("solid", "dashed")) + labs(x = "Representational Gestures Present", y = "Narrative Score") + facet_wrap(~ GroupStatus, scales = "free_x") + theme_bw() + theme(text = element_text(family = "Times New Roman", size = 12), axis.title = element_text(size = 12), axis.text = element_text(size = 10), legend.position = "none", panel.grid.major = element_blank(), panel.grid.minor = element_blank(), panel.border = element_rect(color = "black"), strip.background = element_rect(fill = "white"), strip.text = element_text(size = 12))
```

Total Gestures Scatter Plot with facet_grid

#| label: fig-total-gestures-grid #| fig-cap: "Relationship between total gesture count and narrative scores, faceted by group status and the presence of representational gesture. The top row shows children who used representational gestures during pretend play, while the bottom row shows those who did not." #| fig-width: 8 #| fig-height: 6

First, create a factor for `rep_gesture_present` for better labeling

```
completely_merged_data$rep_gesture_factor <- factor(completely_merged_data$rep_gesture_present, levels =  
c(0, 1), labels = c("No Representational", "Representational"))
```

```
ggplot(completely_merged_data, aes(x = total_gestures, y = AvgHighScore, color = GroupStatus, shape =  
GroupStatus)) + geom_point(size = 3, alpha = 0.8) + geom_smooth(aes(linetype = GroupStatus), method =  
"lm", se = TRUE) + labs(x = "Total Number of Gestures", y = "Narrative Score") + scale_color_grey(start  
= 0.2, end = 0.8) + scale_shape_manual(values = c(16, 17)) + scale_linetype_manual(values =  
c("solid", "dashed")) + # Add facet_grid with rep_gesture_factor in rows and GroupStatus in columns  
facet_grid(rep_gesture_factor ~ GroupStatus, scales = "free_x") + theme_bw() + theme(text =  
element_text(family = "Times New Roman", size = 12), axis.title = element_text(size = 12), axis.text  
= element_text(size = 10), legend.position = "bottom", legend.title = element_blank(), panel.grid.major  
= element_blank(), panel.grid.minor = element_blank(), panel.border = element_rect(color = "black"),  
strip.background = element_rect(fill = "white"), strip.text = element_text(size = 11))
```