

# AllYears\_PP\_Narr\_Analysis

Kristen Johnson

2025-04-24

## Load in packages

```
# 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(effects) # 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

library(broom)
library(gt)

```

## Read in each dataset and add Year column

```

completely_merged_data_H10 <- read_csv(here("PN_DataAnalysis", "PP_Narrative_Analysis", "completely_merger_H10.csv"))

## Rows: 28 Columns: 23
## -- Column specification -----
## Delimiter: ","
## chr  (1): groupstatus
## dbl (21): participant_id, total_gestures, rep_gestures, total_pretend_episod...
## lgl  (1): mlu_missing_flag
##
## 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.
completely_merged_data_H10$Year <- 10

completely_merged_data_H8 <- read_csv(here("PN_DataAnalysis", "PP_Narrative_Analysis", "completely_merger_H8.csv"))

## Rows: 23 Columns: 23

```

```
## -- Column specification -----
## Delimiter: ","
## chr (1): groupstatus
## dbl (21): participant_id, total_gestures, rep_gestures, total_pretend_episod...
## lgl (1): mlu_missing_flag
##
## 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.
completely_merged_data_H8$Year <- 8

completely_merged_data_H7 <- read_csv(here("PN_DataAnalysis", "PP_Narrative_Analysis", "completely_merg...

## Rows: 29 Columns: 23
## -- Column specification -----
## Delimiter: ","
## chr (1): groupstatus
## dbl (21): participant_id, total_gestures, rep_gestures, total_pretend_episod...
## lgl (1): mlu_missing_flag
##
## 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.
completely_merged_data_H7$Year <- 7
```

## Combine data across ALL THREE YEARS

```
# Combine all datasets into one data frame
completely_merged_data_ALLyears <- bind_rows(completely_merged_data_H10, completely_merged_data_H8, comp...

# View the combined dataset
head(completely_merged_data_ALLyears)

## # A tibble: 6 x 24
##   participant_id groupstatus total_gestures rep_gestures total_pretend_episodes
##           <dbl> <chr>           <dbl>           <dbl>           <dbl>
## 1             22 TD                12              1             112
## 2             25 TD                 2              2              21
## 3             28 TD                 7              1              17
## 4             29 TD             100             26             326
## 5             37 TD                 0              0               4
## 6             38 TD             10              2             28
## # i 19 more variables: episodes_with_gesture <dbl>,
## #   prop_episodes_with_gesture <dbl>, prop_rep <dbl>, mlu <dbl>,
## #   mlu_missing_flag <lgl>, rep_gesture_present <dbl>, Project <dbl>,
## #   FoxHOS07 <dbl>, AliceHOS07 <dbl>, AlanHOS07 <dbl>, FoxHOS08 <dbl>,
## #   AlanHOS08 <dbl>, FoxHOS09 <dbl>, AlanHOS09 <dbl>, avg07 <dbl>, avg08 <dbl>,
## #   avg09 <dbl>, max_avg <dbl>, Year <dbl>
```

## run linear regression model (mixed effects wouldn't work)

```
modell1 <- lm(max_avg ~ groupstatus + total_gestures + total_pretend_episodes + mlu + groupstatus:total_g...
              data = completely_merged_data_ALLyears)
```

```
# Model 2: Representational gesture presence as predictor
```

```
model2 <- lm(max_avg ~ groupstatus + rep_gesture_present + total_pretend_episodes + mlu + groupstatus:rep_gesture_present,
             data = completely_merged_data_ALLyears)
```

```
summary(model1)
```

```
##
## Call:
## lm(formula = max_avg ~ groupstatus + total_gestures + total_pretend_episodes +
##     mlu + groupstatus:total_gestures, data = completely_merged_data_ALLyears)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.6906 -1.3752 -0.3281  1.1465  5.1464
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.5534099   0.7589893   4.682 1.26e-05 ***
## groupstatusTD    1.0447204   0.6440451   1.622  0.1090
## total_gestures    0.0338805   0.0349283   0.970  0.3352
## total_pretend_episodes 0.0005178   0.0032013   0.162  0.8719
## mlu              0.3893133   0.1951703   1.995  0.0498 *
## groupstatusTD:total_gestures -0.0516185   0.0348477  -1.481  0.1428
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.109 on 74 degrees of freedom
## Multiple R-squared:  0.1208, Adjusted R-squared:  0.06138
## F-statistic: 2.033 on 5 and 74 DF,  p-value: 0.08372
```

```
summary(model2)
```

```
##
## Call:
## lm(formula = max_avg ~ groupstatus + rep_gesture_present + total_pretend_episodes +
##     mlu + groupstatus:rep_gesture_present, data = completely_merged_data_ALLyears)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.2185 -1.3839 -0.1374  1.1322  5.4587
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.6030978   0.7334269   4.913 5.23e-06 ***
## groupstatusTD    1.2831868   0.6360759   2.017  0.0473 *
## rep_gesture_present 2.5508335   1.1074465   2.303  0.0241 *
## total_pretend_episodes -0.0008164   0.0024188  -0.337  0.7367
## mlu              0.3536237   0.1927513   1.835  0.0706 .
## groupstatusTD:rep_gesture_present -2.9653508   1.2054241  -2.460  0.0162 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.059 on 74 degrees of freedom
## Multiple R-squared:  0.1624, Adjusted R-squared:  0.1059
```

```
## F-statistic: 2.871 on 5 and 74 DF, p-value: 0.02008
```

```
# Calculate effect sizes
```

```
eta_squared(model1)
```

```
## # Effect Size for ANOVA (Type I)
```

```
##
```

```
## Parameter | Eta2 (partial) | 95% CI
```

```
## -----
```

```
## groupstatus | 0.04 | [0.00, 1.00]
```

```
## total_gestures | 1.62e-05 | [0.00, 1.00]
```

```
## total_pretend_episodes | 3.56e-03 | [0.00, 1.00]
```

```
## mlu | 0.06 | [0.00, 1.00]
```

```
## groupstatus:total_gestures | 0.03 | [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 | 0.04 | [0.00, 1.00]
```

```
## rep_gesture_present | 6.42e-03 | [0.00, 1.00]
```

```
## total_pretend_episodes | 5.64e-04 | [0.00, 1.00]
```

```
## mlu | 0.06 | [0.00, 1.00]
```

```
## groupstatus:rep_gesture_present | 0.08 | [0.01, 1.00]
```

```
##
```

```
## - One-sided CIs: upper bound fixed at [1.00].
```

## Make a clean regression table for Model 2:

```
# Turn model output into a tidy table
```

```
model2_table <- tidy(model2) %>%
```

```
  select(term, estimate, std.error, statistic, p.value)
```

```
# Create a nice regression table
```

```
model2_table %>%
```

```
  gt() %>%
```

```
  fmt_number(columns = vars(estimate, std.error, statistic, p.value), decimals = 3) %>%
```

```
  tab_header(title = "Regression Results for Narrative Structure Model")
```

```
## Warning: Since gt v0.3.0, `columns = vars(...)` has been deprecated.
```

```
## * Please use `columns = c(...)` instead.
```

```
#####APA style regression table:
```

```
#colnames(model2)
```

```
# Create APA-style table
```

```
#model2_table %>%
```

```
  # gt() %>%
```

```
  # tab_header(
```

```
    # title = "Table X\nLinear Regression Predicting Narrative Structure Scores"
```

## Regression Results for Narrative Structure Model

term	estimate	std.error	statistic	p.value
(Intercept)	3.603	0.733	4.913	0.000
groupstatusTD	1.283	0.636	2.017	0.047
rep_gesture_present	2.551	1.107	2.303	0.024
total_pretend_episodes	-0.001	0.002	-0.337	0.737
mlu	0.354	0.193	1.835	0.071
groupstatusTD:rep_gesture_present	-2.965	1.205	-2.460	0.016

```
# ) %>%
# cols_label(
#   Term = "Predictor",
#   b = "b",
#   SE = "SE",
#   t = "t",
#   p = "p"
# ) %>%
# tab_options(
#   table.font.size = "small",
#   data_row.padding = px(2),
#   table.align = "center"
# )

library(broom)
library(dplyr)
library(gt)

# Step 1: Prepare tidy model results
model2_table <- tidy(model2) %>%
  select(term, estimate, std.error, statistic, p.value) %>%
  mutate(
    sig = ifelse(p.value < 0.05, TRUE, FALSE), # Create a flag for significance
    p.value = ifelse(p.value < .001, "< .001", sprintf("%.3f", p.value)), # Format p
    estimate = sprintf("%.2f", estimate), # Format estimates
    std.error = sprintf("%.2f", std.error),
    statistic = sprintf("%.2f", statistic)
  ) %>%
  rename(
    Predictor = term,
    b = estimate,
    SE = std.error,
    t = statistic,
    p = p.value
  )

# Step 2: Build the table
model2_table %>%
  gt() %>%
  tab_header(
    title = "Table X",
```

Table X  
Linear Regression Predicting Narrative Structure Scores

Predictor	b	SE	t	p	sig
<b>(Intercept)</b>	<b>3.60</b>	<b>0.73</b>	<b>4.91</b>	<b>&lt; .001</b>	<b>TRUE</b>
<b>groupstatusTD</b>	<b>1.28</b>	<b>0.64</b>	<b>2.02</b>	<b>0.047</b>	<b>TRUE</b>
<b>rep_gesture_present</b>	<b>2.55</b>	<b>1.11</b>	<b>2.30</b>	<b>0.024</b>	<b>TRUE</b>
total_pretend_episodes	-0.00	0.00	-0.34	0.737	FALSE
mlu	0.35	0.19	1.83	0.071	FALSE
<b>groupstatusTD:rep_gesture_present</b>	<b>-2.97</b>	<b>1.21</b>	<b>-2.46</b>	<b>0.016</b>	<b>TRUE</b>

Note. b = unstandardized regression coefficient; SE = standard error; p = significance level. Significant predictors are bolded.

```

    subtitle = "Linear Regression Predicting Narrative Structure Scores"
  ) %>%
  cols_label(
    Predictor = "Predictor",
    b = "b",
    SE = "SE",
    t = "t",
    p = "p"
  ) %>%
  tab_style(
    style = cell_text(weight = "bold"),
    locations = cells_body(
      columns = everything(),
      rows = sig == TRUE
    )
  ) %>%
  tab_source_note(
    source_note = "Note. b = unstandardized regression coefficient; SE = standard error; p = significance level."
  ) %>%
  tab_options(
    table.font.size = "small",
    table.align = "center",
    data_row.padding = px(2)
  )

```

interaction plot

```

library(ggplot2)

# Create a new variable to label group and rep_gesture_present combinations
completely_merged_data_ALLyears$group_rep <- interaction(
  completely_merged_data_ALLyears$groupstatus,
  completely_merged_data_ALLyears$rep_gesture_present,
  sep = " - GesturePresent: "
)

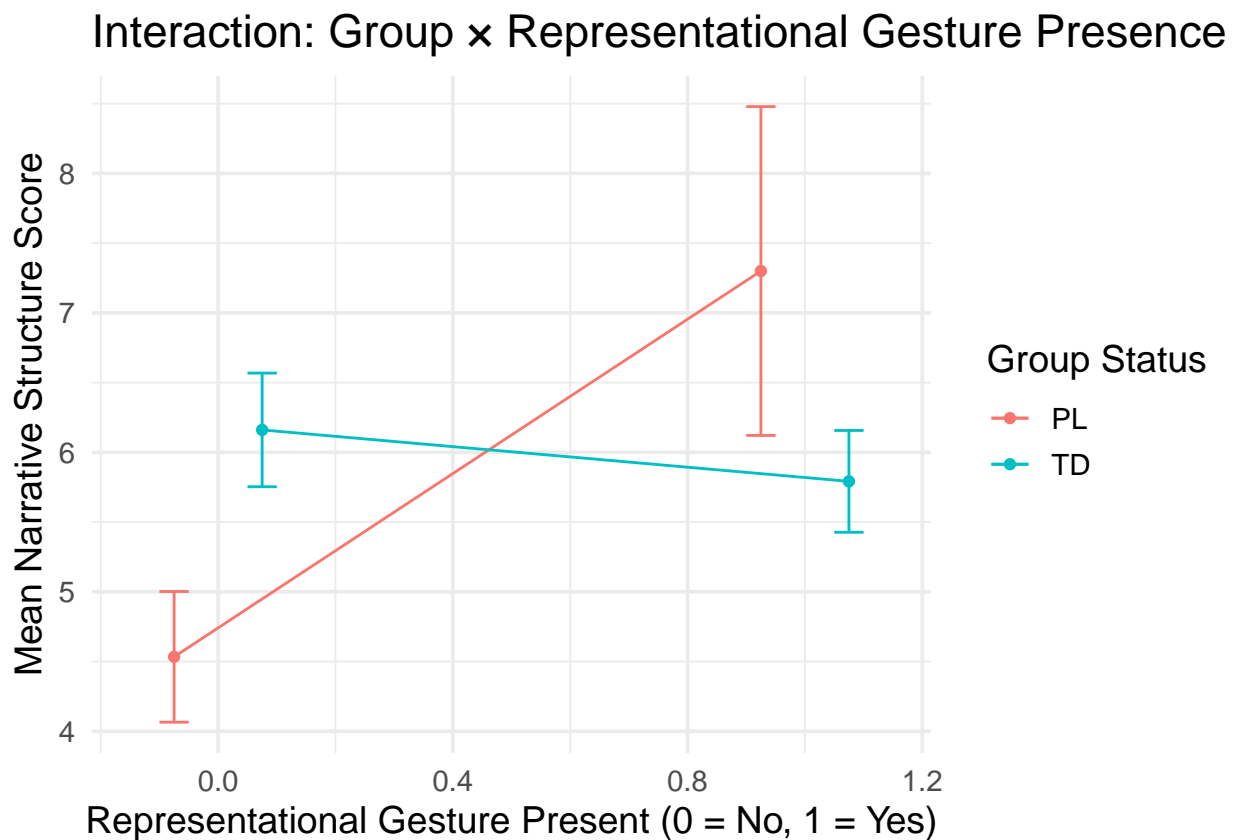
# Create the plot
ggplot(completely_merged_data_ALLyears, aes(x = rep_gesture_present, y = max_avg, color = groupstatus))
  stat_summary(fun = mean, geom = "point", position = position_dodge(width = 0.3)) +

```

```

stat_summary(fun = mean, geom = "line", aes(group = groupstatus), position = position_dodge(width = 0.1),
stat_summary(fun.data = mean_se, geom = "errorbar", width = 0.1, position = position_dodge(width = 0.1),
labs(
  title = "Interaction: Group × Representational Gesture Presence",
  x = "Representational Gesture Present (0 = No, 1 = Yes)",
  y = "Mean Narrative Structure Score",
  color = "Group Status"
) +
theme_minimal(base_size = 14)

```



## scatterplot and facets

```

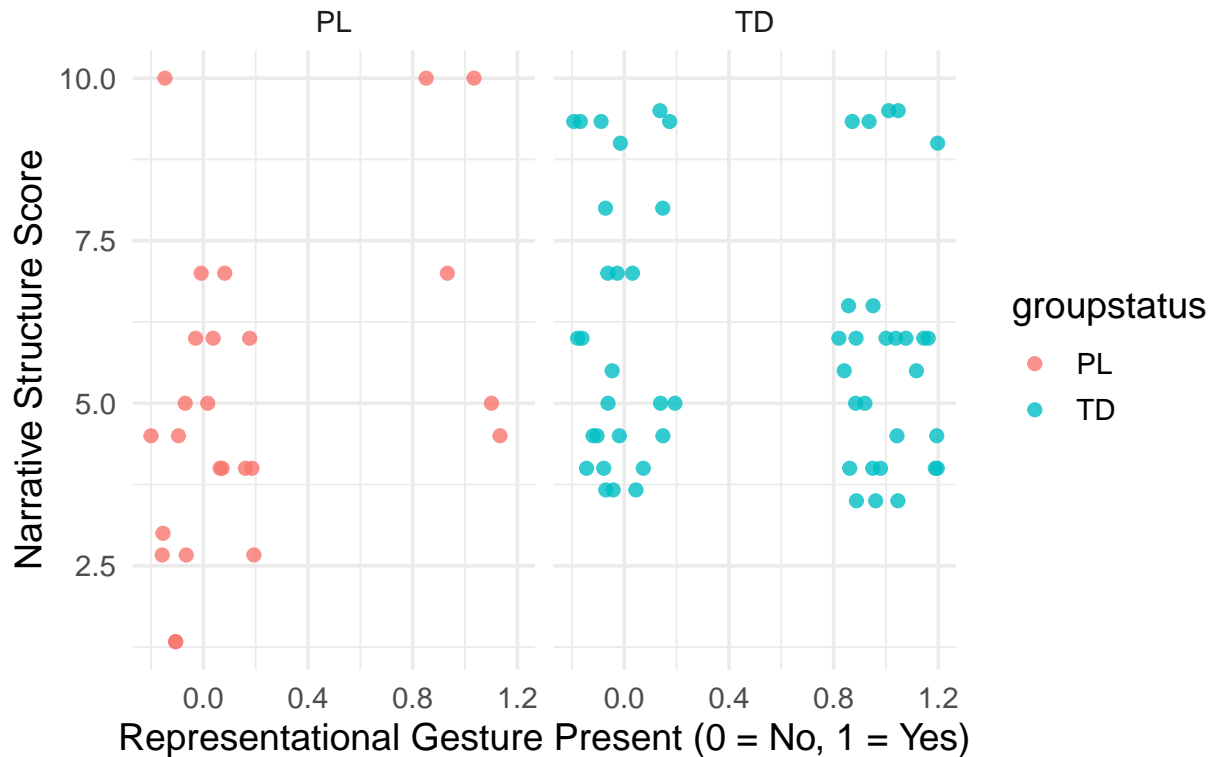
library(ggplot2)

ggplot(completely_merged_data_ALLyears, aes(x = rep_gesture_present, y = max_avg, color = groupstatus))
  geom_jitter(width = 0.2, height = 0, size = 2, alpha = 0.8) +
  facet_wrap(~ groupstatus) +
  labs(
    title = "Narrative Structure Scores by Gesture Presence and Group",
    x = "Representational Gesture Present (0 = No, 1 = Yes)",
    y = "Narrative Structure Score"
  ) +
  theme_minimal(base_size = 14)

```



## Narrative Structure Scores by Gesture Presence and Group



## APA style plot of group $\times$ gesture interaction slopes

```
library(ggplot2)
library(dplyr)

# Create a summary dataset first (means and standard errors)
summary_data <- completely_merged_data_ALLyears %>%
  group_by(groupstatus, rep_gesture_present) %>%
  summarize(
    mean_score = mean(max_avg, na.rm = TRUE),
    se_score = sd(max_avg, na.rm = TRUE) / sqrt(n())
  )
```

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

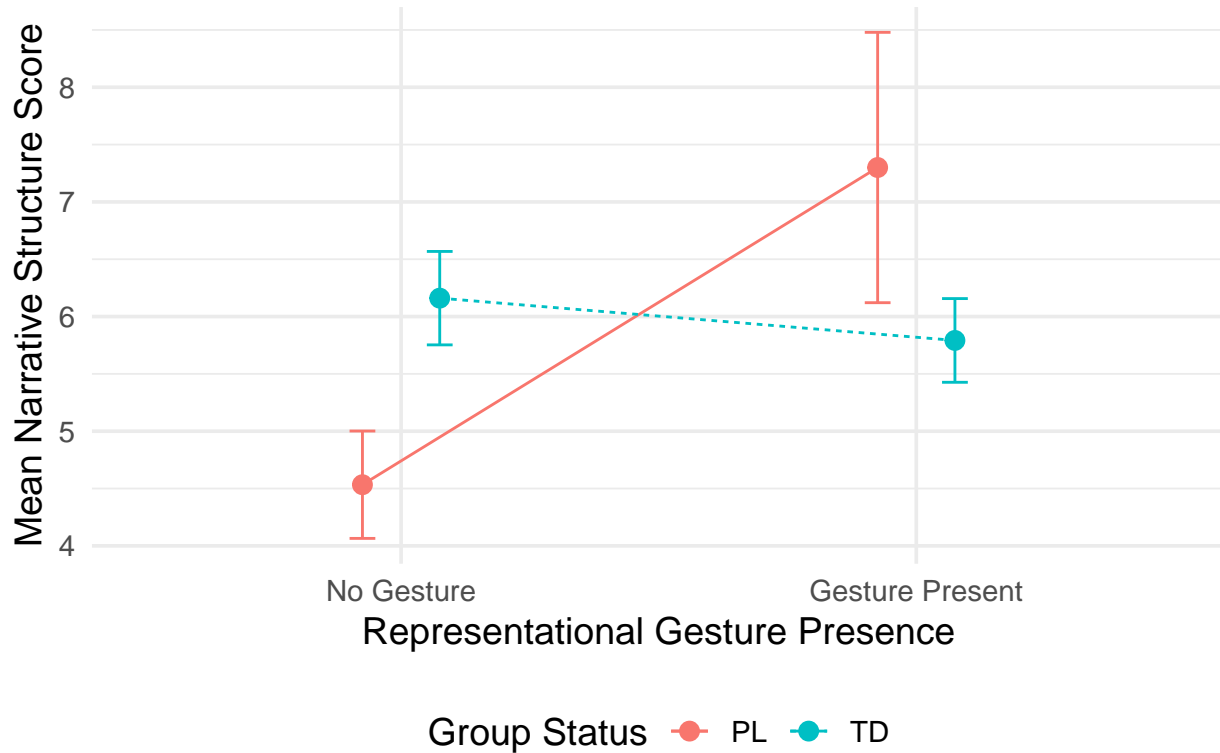
```
# Plot
ggplot(summary_data, aes(x = factor(rep_gesture_present), y = mean_score, group = groupstatus, color = groupstatus)) +
  geom_point(position = position_dodge(width = 0.3), size = 3) +
  geom_line(position = position_dodge(width = 0.3), aes(linetype = groupstatus)) +
  geom_errorbar(aes(ymin = mean_score - se_score, ymax = mean_score + se_score),
    width = 0.1, position = position_dodge(width = 0.3)) +
  scale_x_discrete(labels = c("No Gesture", "Gesture Present")) +
  labs(
    title = "Interaction Between Group and Representational Gesture Presence",
    x = "Representational Gesture Presence",
```

```

y = "Mean Narrative Structure Score",
color = "Group Status",
linetype = "Group Status"
) +
theme_minimal(base_size = 14) +
theme(
  legend.position = "bottom",
  plot.title = element_text(hjust = 0.5)
)

```

## Interaction Between Group and Representational Gesture Presence



#with overlay of individuals

```

library(ggplot2)
library(dplyr)

# Step 1: Summary dataset for group means and SE
summary_data <- completely_merged_data_ALLyears %>%
  group_by(groupstatus, rep_gesture_present) %>%
  summarize(
    mean_score = mean(max_avg, na.rm = TRUE),
    se_score = sd(max_avg, na.rm = TRUE) / sqrt(n()),
    .groups = "drop"
  )

# Step 2: Plot
ggplot() +
  # 2a: Individual participant points (light color)
  geom_jitter(

```

```

data = completely_merged_data_ALLyears,
aes(x = factor(rep_gesture_present), y = max_avg, color = groupstatus),
width = 0.2, height = 0, alpha = 0.4, size = 2
) +

# 2b: Group means (strong color)
geom_point(
  data = summary_data,
  aes(x = factor(rep_gesture_present), y = mean_score, color = groupstatus),
  position = position_dodge(width = 0.3),
  size = 4
) +

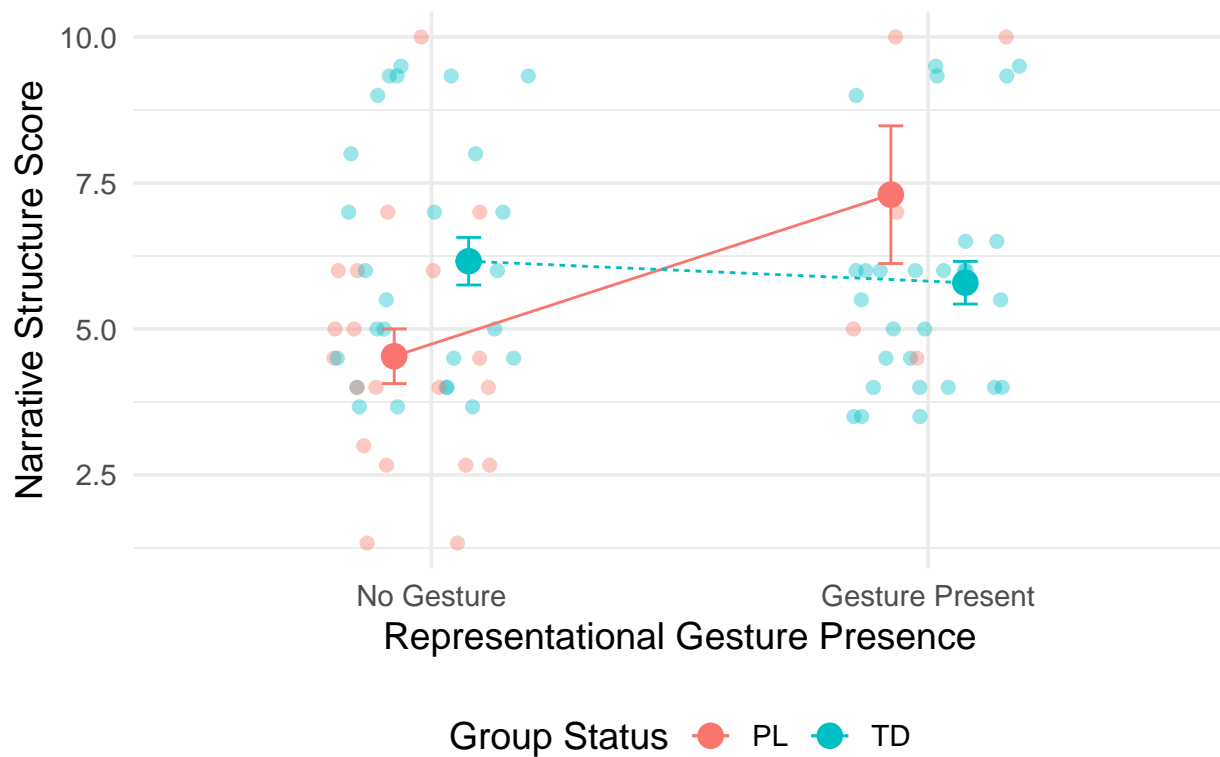
# 2c: Lines connecting group means
geom_line(
  data = summary_data,
  aes(x = factor(rep_gesture_present), y = mean_score, group = groupstatus, color = groupstatus, line
  position = position_dodge(width = 0.3)
) +

# 2d: Error bars for means
geom_errorbar(
  data = summary_data,
  aes(x = factor(rep_gesture_present), ymin = mean_score - se_score, ymax = mean_score + se_score, co
  width = 0.1,
  position = position_dodge(width = 0.3)
) +

# 2e: Labels and Themes
scale_x_discrete(labels = c("No Gesture", "Gesture Present")) +
labs(
  title = "Interaction of Group Status and Representational Gesture Presence",
  x = "Representational Gesture Presence",
  y = "Narrative Structure Score",
  color = "Group Status",
  linetype = "Group Status"
) +
theme_minimal(base_size = 14) +
theme(
  legend.position = "bottom",
  plot.title = element_text(hjust = 0.5)
)

```

## Interaction of Group Status and Representational Gesture Presence



## fancy graph

```
library(ggplot2)
library(dplyr)

# Summary dataset again
summary_data <- completely_merged_data_ALLyears %>%
  group_by(groupstatus, rep_gesture_present) %>%
  summarize(
    mean_score = mean(max_avg, na.rm = TRUE),
    se_score = sd(max_avg, na.rm = TRUE) / sqrt(n()),
    .groups = "drop"
  )

# Full plot with light gray background
ggplot() +
  # Individual points (light)
  geom_jitter(
    data = completely_merged_data_ALLyears,
    aes(x = factor(rep_gesture_present), y = max_avg, color = groupstatus),
    width = 0.2, height = 0, alpha = 0.4, size = 2
  ) +
  # Mean points (strong)
  geom_point(
    data = summary_data,
```

```

aes(x = factor(rep_gesture_present), y = mean_score, color = groupstatus),
position = position_dodge(width = 0.3),
size = 4
) +

# Lines between means
geom_line(
  data = summary_data,
  aes(x = factor(rep_gesture_present), y = mean_score, group = groupstatus, color = groupstatus, line
position = position_dodge(width = 0.3)
) +

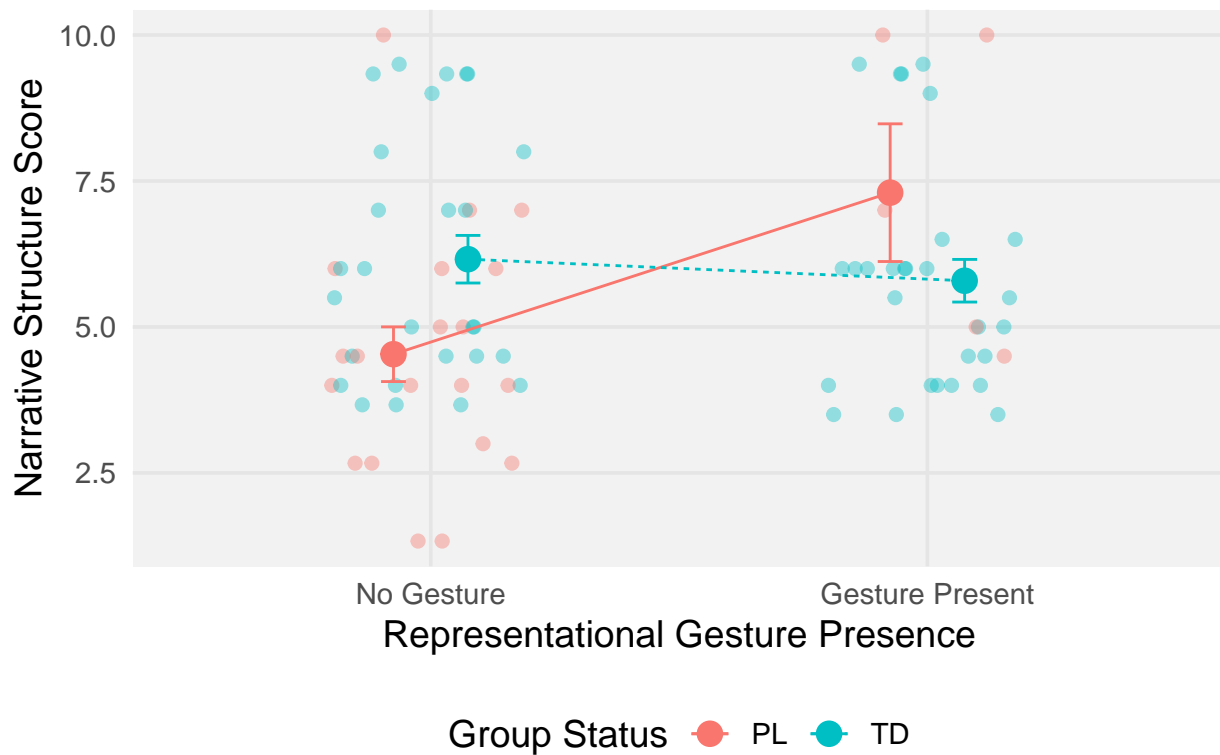
# Error bars
geom_errorbar(
  data = summary_data,
  aes(x = factor(rep_gesture_present), ymin = mean_score - se_score, ymax = mean_score + se_score, col
width = 0.1,
position = position_dodge(width = 0.3)
) +

# Labels
scale_x_discrete(labels = c("No Gesture", "Gesture Present")) +
labs(
  title = "Interaction of Group Status and Representational Gesture Presence",
  x = "Representational Gesture Presence",
  y = "Narrative Structure Score",
  color = "Group Status",
  linetype = "Group Status"
) +

# Themes
theme_minimal(base_size = 14) +
theme(
  panel.background = element_rect(fill = "gray95", color = NA), # Light gray background
  plot.background = element_rect(fill = "white", color = NA),
  panel.grid.major = element_line(color = "gray90"),
  panel.grid.minor = element_blank(),
  legend.position = "bottom",
  plot.title = element_text(hjust = 0.5, face = "bold")
)

```

## Interaction of Group Status and Representational Gesture Presence



```
library(ggplot2)
library(dplyr)

# Step 1: Summarize group means and standard errors
summary_data <- completely_merged_data_ALLyears %>%
  group_by(groupstatus, rep_gesture_present) %>%
  summarize(
    mean_score = mean(max_avg, na.rm = TRUE),
    se_score = sd(max_avg, na.rm = TRUE) / sqrt(n()),
    .groups = "drop"
  )

# Step 2: Plot with ribbons
ggplot() +
  # 2a: Individual points (light gray dots)
  geom_jitter(
    data = completely_merged_data_ALLyears,
    aes(x = factor(rep_gesture_present), y = max_avg, color = groupstatus),
    width = 0.2, height = 0, alpha = 0.4, size = 2
  ) +
  # 2b: Ribbons for SE
  geom_ribbon(
    data = summary_data,
    aes(
      x = as.numeric(factor(rep_gesture_present)),
      ymin = mean_score - se_score,
      ymax = mean_score + se_score,
      fill = groupstatus
    )
  )
```

```

    fill = groupstatus,
    group = groupstatus
  ),
  inherit.aes = FALSE,
  alpha = 0.2
) +

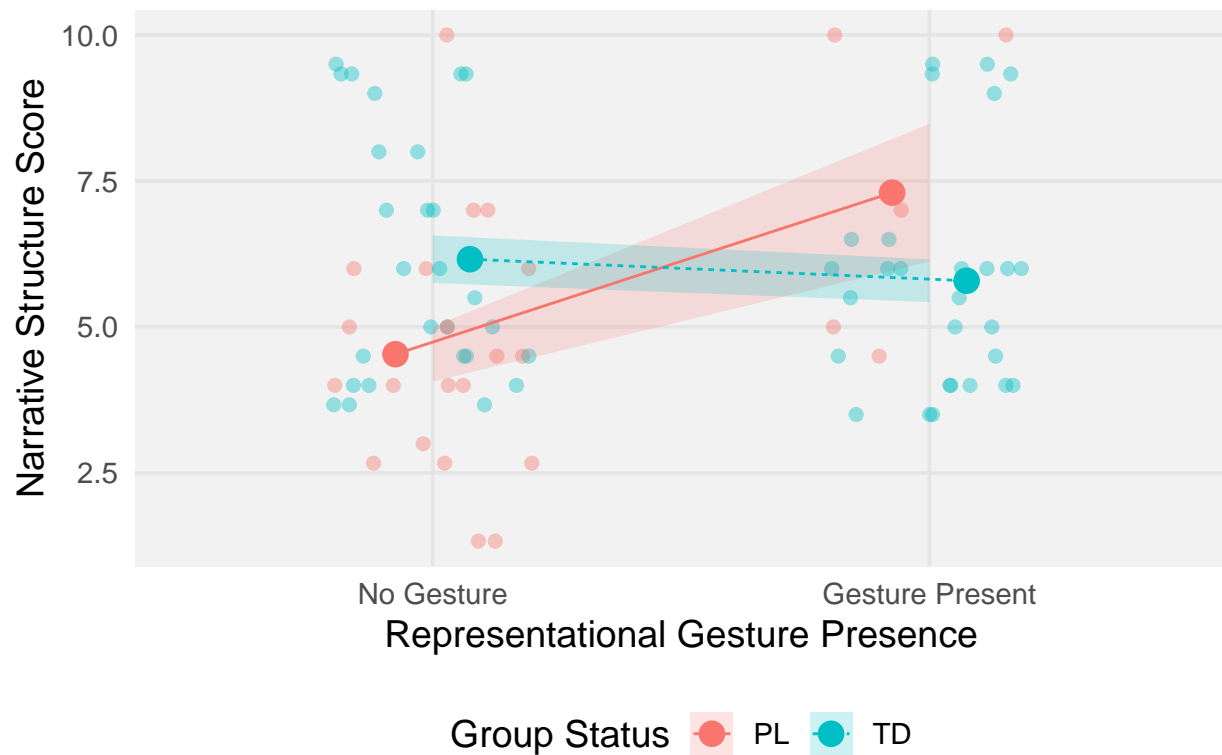
# 2c: Mean points
geom_point(
  data = summary_data,
  aes(x = factor(rep_gesture_present), y = mean_score, color = groupstatus),
  position = position_dodge(width = 0.3),
  size = 4
) +

# 2d: Lines connecting means
geom_line(
  data = summary_data,
  aes(x = factor(rep_gesture_present), y = mean_score, group = groupstatus, color = groupstatus, line
  position = position_dodge(width = 0.3)
) +

# 2e: Labels and themes
scale_x_discrete(labels = c("No Gesture", "Gesture Present")) +
labs(
  title = "Interaction of Group Status and Representational Gesture Presence",
  x = "Representational Gesture Presence",
  y = "Narrative Structure Score",
  color = "Group Status",
  fill = "Group Status",
  linetype = "Group Status"
) +
theme_minimal(base_size = 14) +
theme(
  panel.background = element_rect(fill = "gray95", color = NA),
  plot.background = element_rect(fill = "white", color = NA),
  panel.grid.major = element_line(color = "gray90"),
  panel.grid.minor = element_blank(),
  legend.position = "bottom",
  plot.title = element_text(hjust = 0.5, face = "bold")
)

```

## Interaction of Group Status and Representational Gesture Presence



## MIXED MODEL & TROUBLESHOOTING

```
# Model 1: Total gestures as predictor
mixed_model1 <- lmer(max_avg ~ groupstatus + total_gestures + total_pretend_episodes + mlu + groupstatus
  data = completely_merged_data_ALLyears)

# Model 2: Representational gesture presence as predictor
mixed_model2 <- lmer(max_avg ~ groupstatus + rep_gesture_present + total_pretend_episodes + mlu + groupstatus
  data = completely_merged_data_ALLyears)

summary(mixed_model1)
summary(mixed_model2)

# Calculate effect sizes
eta_squared(mixed_model1)
eta_squared(mixed_model2)

completely_merged_data_ALLyears <- completely_merged_data_ALLyears %>%
  mutate(
    total_gestures_z = scale(total_gestures),
    total_pretend_episodes_z = scale(total_pretend_episodes),
    mlu_z = scale(mlu),
    prop_episodes_with_gesture_z = scale(prop_episodes_with_gesture)
  )

library(lme4)
```



```

mixed_model1_scaled <- lmer(max_avg ~ groupstatus + total_gestures_z + total_pretend_episodes_z + mlu_z +
  groupstatus:total_gestures_z + (1 | participant_id),
  data = completely_merged_data_ALLyears,
  control = lmerControl(optimizer = "bobyqa"))

glimpse(completely_merged_data_ALLyears)

completely_merged_data_ALLyears$total_gestures_z <- as.numeric(scale(completely_merged_data_ALLyears$total_gestures_z))
completely_merged_data_ALLyears$total_pretend_episodes_z <- as.numeric(scale(completely_merged_data_ALLyears$total_pretend_episodes_z))
completely_merged_data_ALLyears$mlu_z <- as.numeric(scale(completely_merged_data_ALLyears$mlu_z))

completely_merged_data_ALLyears %>%
  select(total_gestures, total_pretend_episodes, mlu, max_avg) %>%
  cor(use = "complete.obs")

model1_z <- lmer(max_avg ~ groupstatus + total_gestures_z + total_pretend_episodes_z + mlu_z + groupstatus:total_gestures_z +
  data = completely_merged_data_ALLyears)
summary(model1_z)

model1 <- lmer(
  max_avg ~ total_gestures_z + total_pretend_episodes_z + mlu_z +
  (1 | participant_id),
  data = completely_merged_data_ALLyears,
  control = lmerControl(optimizer = "bobyqa")
)
summary(model1)

# Using proportion of episodes with gestures
model3 <- lmer(
  max_avg ~ prop_episodes_with_gesture_z + mlu_z +
  (1 | participant_id),
  data = completely_merged_data_ALLyears,
  control = lmerControl(optimizer = "bobyqa")
)
summary(model3)

cor(completely_merged_data_ALLyears$prop_episodes_with_gesture_z,
  completely_merged_data_ALLyears$mlu_z, use = "complete.obs")

vif(lm(max_avg ~ prop_episodes_with_gesture_z + mlu_z,
  data = completely_merged_data_ALLyears))

# Boxplot for prop_episodes_with_gesture_z
boxplot(completely_merged_data_ALLyears$prop_episodes_with_gesture_z, main = "Boxplot of prop_episodes_with_gesture_z")

# Boxplot for mlu_z
boxplot(completely_merged_data_ALLyears$mlu_z, main = "Boxplot of mlu_z")

# Calculate Z-scores for prop_episodes_with_gesture_z and mlu_z
z_scores <- scale(completely_merged_data_ALLyears[, c("prop_episodes_with_gesture_z", "mlu_z")])

# Identify outliers with Z-scores greater than 3 or less than -3
outliers <- which(abs(z_scores) > 3, arr.ind = TRUE)
outliers

```

```

# Identify the rows
completely_merged_data_ALLyears[c(17, 48), c("participant_id", "prop_episodes_with_gesture_z", "prop_ep

# Install if needed
install.packages("robustlmm")
library(robustlmm)

model3_robust <- rlmer(
  max_avg ~ prop_episodes_with_gesture_z + mlu_z + (1 | participant_id),
  data = completely_merged_data_ALLyears
)

summary(model3_robust)

# Exclude participants 75 and 117
cleaned_data <- completely_merged_data_ALLyears %>%
  filter(!participant_id %in% c(75, 117))

# Refit the mixed model
model3_cleaned <- lmer(
  max_avg ~ prop_episodes_with_gesture_z + mlu_z + (1 | participant_id),
  data = cleaned_data,
  control = lmerControl(optimizer = "bobyqa")
)

# View results
summary(model3_cleaned)

VarCorr(model3_cleaned)

```