

AllYears_PP_Narr_Analysis

Kristen Johnson

2025-04-24

Setup

```
##Load in packages
```

```
## -- 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
## 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
##
## Loading required package: zoo
##
##
## Attaching package: 'zoo'
##
##
## The following objects are masked from 'package:base':
##
##   as.Date, as.Date.numeric
##
## here() starts at /Users/kristenjohnson/KristenWorkingDirectory/Play_Narrative
##
```

```
## Loading required package: Matrix
##
##
## Attaching package: 'Matrix'
##
##
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
##
##
## Attaching package: 'lmerTest'
##
##
## The following object is masked from 'package:lme4':
##
##     lmer
##
## The following object is masked from 'package:stats':
##
##     step
##
## Registering fonts with R
##
##
## Attaching package: 'boot'
##
##
## The following object is masked from 'package:car':
##
##     logit
```

Read in each dataset and add Year column

```
completely_merged_data_H10 <- read_csv(here("PN_DataAnalysis", "PP_Narrative_Analysis", "completely_merg

## 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_merg

## 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>

# write a csv
write_csv(completely_merged_data_ALLyears, here("PN_DataAnalysis", "PP_Narrative_Analysis", "completely_r...

#OLS regressions ## Linear regression Models 1 & 2 (mixed effects wouldn't work)- OLS
modell_ols <- lm(max_avg ~ groupstatus + total_gestures + total_pretend_episodes + mlu + Year + groupsta...
data = completely_merged_data_ALLyears)
```

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

```
model2_ols <- lm(max_avg ~ groupstatus + rep_gesture_present + total_pretend_episodes + mlu + Year + gr
                data = completely_merged_data_ALLyears)
```

```
summary(model1_ols)
```

```
##
## Call:
## lm(formula = max_avg ~ groupstatus + total_gestures + total_pretend_episodes +
##     mlu + Year + groupstatus:total_gestures, data = completely_merged_data_ALLyears)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.7049 -1.3657 -0.3359  1.1299  5.1655
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      3.6474436   1.6690423    2.185  0.0321 *
## groupstatusTD      1.0400143   0.6526622    1.593  0.1154
## total_gestures      0.0335295   0.0355993    0.942  0.3494
## total_pretend_episodes 0.0005231  0.0032241    0.162  0.8716
## mlu                0.3929242   0.2045919    1.921  0.0587 .
## Year             -0.0123314   0.1945876   -0.063  0.9496
## groupstatusTD:total_gestures -0.0513391  0.0353604   -1.452  0.1508
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.124 on 73 degrees of freedom
## Multiple R-squared:  0.1208, Adjusted R-squared:  0.04857
## F-statistic: 1.672 on 6 and 73 DF,  p-value: 0.14
```

```
summary(model2_ols)
```

```
##
## Call:
## lm(formula = max_avg ~ groupstatus + rep_gesture_present + total_pretend_episodes +
##     mlu + Year + groupstatus:rep_gesture_present, data = completely_merged_data_ALLyears)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.1786 -1.3842 -0.1442  1.1078  5.4079
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      3.3597716   1.6256847    2.067  0.0423 *
## groupstatusTD      1.2980079   0.6463431    2.008  0.0483 *
## rep_gesture_present  2.5758356   1.1246803    2.290  0.0249 *
## total_pretend_episodes -0.0007954  0.0024381   -0.326  0.7452
## mlu                0.3442223   0.2019382    1.705  0.0925 .
## Year                0.0319182   0.1899892    0.168  0.8670
## groupstatusTD:rep_gesture_present -2.9979707  1.2288544   -2.440  0.0171 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 2.072 on 73 degrees of freedom
## Multiple R-squared:  0.1628, Adjusted R-squared:  0.09396
## F-statistic: 2.365 on 6 and 73 DF,  p-value: 0.03829
```

```
# Calculate effect sizes
```

```
eta_squared(model1_ols)
```

```
## # 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]
```

```
## Year | 8.30e-04 | [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_ols)
```

```
## # 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]
```

```
## Year | 6.64e-04 | [0.00, 1.00]
```

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

```
##
```

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

Combined regression table (OLS) of Model 1 and Model 2

```
library(broom)
```

```
library(dplyr)
```

```
library(gt)
```

```
# Step 1: Tidy and format original OLS models
```

```
# Model 1 (Total Gestures)
```

```
model1_ols_tidy <- tidy(model1_ols) %>%
```

```
  mutate(
```

```
    Predictor = case_when(
```

```
      term == "(Intercept)" ~ "Intercept",
```

```
      term == "groupstatusTD" ~ "Group Status (TD vs PL)",
```

```
      term == "total_gestures" ~ "Total Gestures",
```

```
      term == "total_pretend_episodes" ~ "Total Pretend Episodes",
```

```
      term == "mlu" ~ "Mean Length of Utterance (MLU)",
```

```
      term == "groupstatusTD:total_gestures" ~ "Group × Total Gestures Interaction",
```

```
      TRUE ~ term
```

```
)
```

```

) %>%
select(Predictor, estimate, std.error, p.value)

# Model 2 (Gesture Presence)
model2_ols_tidy <- tidy(model2_ols) %>%
mutate(
  Predictor = case_when(
    term == "(Intercept)" ~ "Intercept",
    term == "groupstatusTD" ~ "Group Status (TD vs PL)",
    term == "rep_gesture_present" ~ "Gesture Presence",
    term == "total_pretend_episodes" ~ "Total Pretend Episodes",
    term == "mlu" ~ "Mean Length of Utterance (MLU)",
    term == "groupstatusTD:rep_gesture_present" ~ "Group x Gesture Presence Interaction",
    TRUE ~ term
  )
) %>%
select(Predictor, estimate, std.error, p.value)

# Step 2: Full join on Predictor
combined_ols_table <- full_join(model1_ols_tidy, model2_ols_tidy, by = "Predictor", suffix = c("_Model1", "_Model2"))
mutate(
  Sig1 = case_when(
    p.value_Model1 < .001 ~ "***",
    p.value_Model1 < .01 ~ "**",
    p.value_Model1 < .05 ~ "*",
    TRUE ~ ""
  ),
  Sig2 = case_when(
    p.value_Model2 < .001 ~ "***",
    p.value_Model2 < .01 ~ "**",
    p.value_Model2 < .05 ~ "*",
    TRUE ~ ""
  ),
  `b (Model 1)` = ifelse(!is.na(estimate_Model1), sprintf("%.2f", estimate_Model1), NA),
  `SE (Model 1)` = ifelse(!is.na(std.error_Model1), sprintf("%.2f", std.error_Model1), NA),
  `b (Model 2)` = ifelse(!is.na(estimate_Model2), sprintf("%.2f", estimate_Model2), NA),
  `SE (Model 2)` = ifelse(!is.na(std.error_Model2), sprintf("%.2f", std.error_Model2), NA),
  `b (Model 1)` = paste0(`b (Model 1)`, Sig1),
  `b (Model 2)` = paste0(`b (Model 2)`, Sig2)
) %>%
select(Predictor, `b (Model 1)`, `SE (Model 1)`, `b (Model 2)`, `SE (Model 2)`, p.value_Model1, p.value_Model2)

# Step 3: Make the gt table
combined_ols_table %>%
gt() %>%
tab_header(
  title = "Table X",
  subtitle = "OLS Regression Predicting Narrative Structure Scores Across Models"
) %>%
cols_label(
  Predictor = "Predictor",
  `b (Model 1)` = "b (Model 1)",
  `SE (Model 1)` = "SE (Model 1)",

```

Table X
OLS Regression Predicting Narrative Structure Scores Across Models

Predictor	b (Model 1)	SE (Model 1)	b (Model 2)	SE (Model 2)
Intercept	3.65*	1.67	3.36*	1.63
Group Status (TD vs PL)	1.04	0.65	1.30*	0.65
Total Gestures	0.03	0.04	NA	NA
Total Pretend Episodes	0.00	0.00	-0.00	0.00
Mean Length of Utterance (MLU)	0.39	0.20	0.34	0.20
Year	-0.01	0.19	0.03	0.19
Group × Total Gestures Interaction	-0.05	0.04	NA	NA
Gesture Presence	NA	NA	2.58*	1.12
Group × Gesture Presence Interaction	NA	NA	-3.00*	1.23

Note. b = unstandardized regression coefficient; SE = standard error. *p* < .05, **p** < .01, ***p*** < .001.

```

`b (Model 2)` = "b (Model 2)",
`SE (Model 2)` = "SE (Model 2)"
) %>%
tab_style(
  style = cell_text(weight = "bold"),
  locations = cells_body(
    rows = (p.value_Model1 < 0.05) | (p.value_Model2 < 0.05),
    columns = c(`b (Model 1)`, `b (Model 2)`)
  )
) %>%
cols_hide(columns = c(p.value_Model1, p.value_Model2)) %>%
tab_options(
  table.font.size = "small",
  heading.align = "center",
  data_row.padding = px(2),
  column_labels.font.weight = "bold",
  table.width = pct(90)
) %>%
tab_source_note(
  source_note = "Note. b = unstandardized regression coefficient; SE = standard error. *p* < .05, **p** < .01, ***p*** < .001."
)

```

Reduced Versions of Model 2 (OLS), dropping nonsignificant predictors

```

# Lean version of Model 2: Representational gesture presence as predictor
model2_ols_lean <- lm(max_avg ~ groupstatus + rep_gesture_present + mlu + Year + groupstatus:rep_gesture_present,
  data = completely_merged_data_ALLyears)

# Even leaner version of Model 2: dropping Year
model2_ols_lean_noyear <- lm(max_avg ~ groupstatus + rep_gesture_present + mlu + groupstatus:rep_gesture_present,
  data = completely_merged_data_ALLyears)

summary(model2_ols_lean)

```

```
##
## Call:
```

```
## lm(formula = max_avg ~ groupstatus + rep_gesture_present + mlu +
##      Year + groupstatus:rep_gesture_present, data = completely_merged_data_ALLyears)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.1884 -1.3935 -0.1724  1.1777  5.3679
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      3.27429      1.59472   2.053  0.0436 *
## groupstatusTD      1.29622      0.64241   2.018  0.0472 *
## rep_gesture_present 2.45874      1.05942   2.321  0.0231 *
## mlu                0.34448      0.20071   1.716  0.0903 .
## Year              0.03509      0.18859   0.186  0.8529
## groupstatusTD:rep_gesture_present -2.91719      1.19636  -2.438  0.0172 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.06 on 74 degrees of freedom
## Multiple R-squared:  0.1615, Adjusted R-squared:  0.1049
## F-statistic: 2.852 on 5 and 74 DF,  p-value: 0.02075
```

```
summary(model2_ols_lean_noyear)
```

```
##
## Call:
## lm(formula = max_avg ~ groupstatus + rep_gesture_present + mlu +
##      groupstatus:rep_gesture_present, data = completely_merged_data_ALLyears)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.2326 -1.4246 -0.1678  1.1964  5.4228
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      3.5400      0.7050   5.021 3.37e-06 ***
## groupstatusTD      1.2798      0.6322   2.024  0.0465 *
## rep_gesture_present 2.4278      1.0395   2.335  0.0222 *
## mlu                0.3548      0.1916   1.852  0.0679 .
## groupstatusTD:rep_gesture_present -2.8789      1.1709  -2.459  0.0163 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.047 on 75 degrees of freedom
## Multiple R-squared:  0.1612, Adjusted R-squared:  0.1164
## F-statistic: 3.602 on 4 and 75 DF,  p-value: 0.009678
```

```
# Calculate effect sizes
eta_squared(model2_ols_lean)
```

```
## # Effect Size for ANOVA (Type I)
##
## Parameter | Eta2 (partial) | 95% CI
## -----|-----|-----
## groupstatus | 0.04 | [0.00, 1.00]
```



```
## rep_gesture_present | 6.41e-03 | [0.00, 1.00]
## mlu | 0.06 | [0.00, 1.00]
## Year | 7.59e-04 | [0.00, 1.00]
## groupstatus:rep_gesture_present | 0.07 | [0.01, 1.00]
##
## - One-sided CIs: upper bound fixed at [1.00].
```

```
eta_squared(model2_ols_lean_noyear)
```

```
## # Effect Size for ANOVA (Type I)
##
## Parameter | Eta2 (partial) | 95% CI
## -----
## groupstatus | 0.04 | [0.00, 1.00]
## rep_gesture_present | 6.41e-03 | [0.00, 1.00]
## mlu | 0.06 | [0.00, 1.00]
## groupstatus:rep_gesture_present | 0.07 | [0.01, 1.00]
##
## - One-sided CIs: upper bound fixed at [1.00].
```

APA Table of Reduced Model 2

```
# Load necessary libraries
```

```
library(broom)
```

```
library(dplyr)
```

```
library(gt)
```

```
# Assuming your reduced model 2 is stored as 'model2_reduced'
```

```
# Example:
```

```
# model2_reduced <- lm(max_avg ~ groupstatus + rep_gesture_present + mlu + groupstatus:rep_gesture_pres
```

```
# Step 1: Tidy the model output
```

```
model2_reduced_table <- tidy(model2_ols_lean_noyear) %>%
  mutate(
    sig = case_when(
      p.value < 0.001 ~ "***",
      p.value < 0.01 ~ "**",
      p.value < 0.05 ~ "*",
      p.value < 0.10 ~ "+",
      TRUE ~ ""
    ),
    b_formatted = sprintf("%.2f", estimate),
    SE_formatted = sprintf("%.2f", std.error),
    t_formatted = sprintf("%.2f", statistic),
    p_formatted = ifelse(p.value < .001, "< .001", sprintf("%.3f", p.value))
  ) %>%
  rename(
    Predictor = term
  ) %>%
  select(Predictor, b_formatted, SE_formatted, t_formatted, p_formatted, sig)
```

```
# Step 2: Create the APA-style gt table
```

```
model2_reduced_table %>%
```

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

Table X
Reduced Linear Regression Predicting Narrative Structure Scores

Predictor	b	SE	t	p	
(Intercept)	3.54	0.70	5.02	< .001	***
groupstatusTD	1.28	0.63	2.02	0.046	*
rep_gesture_present	2.43	1.04	2.34	0.022	*
mlu	0.35	0.19	1.85	0.068	†
groupstatusTD:rep_gesture_present	-2.88	1.17	-2.46	0.016	*

Note. b = unstandardized regression coefficient; SE = standard error; t = t value; p = significance level. Significance indicated by * p < .05, ** p < .01, *** p < .001, † p < .10.

```

tab_header(
  title = "Table X",
  subtitle = "Reduced Linear Regression Predicting Narrative Structure Scores"
) %>%
cols_label(
  Predictor = "Predictor",
  b_formatted = "b",
  SE_formatted = "SE",
  t_formatted = "t",
  p_formatted = "p",
  sig = ""
) %>%
fmt_markdown(columns = vars(Predictor)) %>%
tab_style(
  style = list(
    cell_text(weight = "bold")
  ),
  locations = cells_body(
    rows = sig != "" # Bold significant rows
  )
) %>%
tab_source_note(
  source_note = "Note. b = unstandardized regression coefficient; SE = standard error; t = t value; p"
) %>%
tab_options(
  table.font.size = "small",
  table.align = "center",
  data_row.padding = px(2)
)

```

```

## Warning: Since gt v0.3.0, `columns = vars(...)` has been deprecated.
## * Please use `columns = c(...)` instead.

```

Simple Slope of Predicted Outcomes for Model 2 (probs for appendix)

```

library(ggeffects)

# Create the predicted data frame
effect_df <- ggpredict(model2_ols_lean_noyear, terms = c("rep_gesture_present", "groupstatus"))

```

```

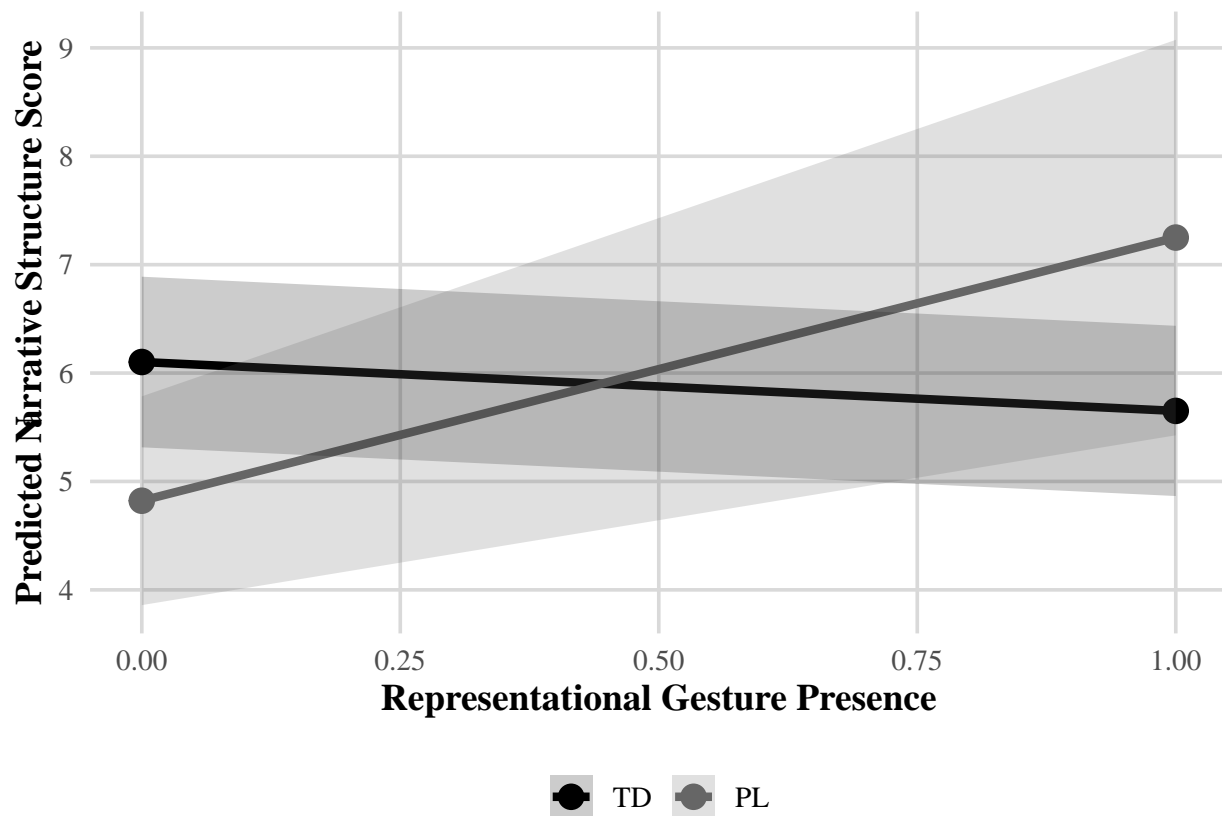
## Some of the focal terms are of type `character`. This may lead to
## unexpected results. It is recommended to convert these variables to
## factors before fitting the model.
## The following variables are of type character: `groupstatus`

# install.packages("extrafont")
library(extrafont)

# APA-style polished plot
ggplot(effect_df, aes(x = x, y = predicted, color = group)) +
  geom_point(size = 4) +
  geom_line(size = 1.5) +
  geom_ribbon(aes(ymin = conf.low, ymax = conf.high, fill = group), alpha = 0.2, color = NA) +
  scale_color_manual(values = c("black", "gray40")) +
  scale_fill_manual(values = c("black", "gray40")) +
  labs(
    title = NULL,
    x = "Representational Gesture Presence",
    y = "Predicted Narrative Structure Score",
    color = "Group",
    fill = "Group"
  ) +
  theme_minimal(base_size = 14, base_family = "Times New Roman") +
  theme(
    legend.position = "bottom",
    plot.title = element_text(face = "bold", hjust = 0.5),
    axis.title = element_text(face = "bold"),
    panel.grid.major = element_line(color = "gray85"),
    panel.grid.minor = element_blank(),
    legend.title = element_blank()
  )
)

## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

```



APA Interaction Plot: Model 2's observed means and standard errors

```
# 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: APA-style Plot without ribbons
ggplot() +

  # 2a: Individual participant points (optional jitter for transparency)
  geom_jitter(
    data = completely_merged_data_ALLyears,
    aes(x = factor(rep_gesture_present), y = max_avg, shape = groupstatus),
    width = 0.2, height = 0, alpha = 0.6, size = 2, color = "black"
  ) +

  # 2b: Vertical error bars ( $\pm 1$  SE)
  geom_errorbar(
    data = summary_data,
    aes(
      x = factor(rep_gesture_present),
      ymin = mean_score - se_score,
```

```

    ymax = mean_score + se_score,
    group = groupstatus
  ),
  width = 0.1,
  position = position_dodge(width = 0.3),
  color = "black",
  size = 0.8
) +

# 2c: Mean points
geom_point(
  data = summary_data,
  aes(x = factor(rep_gesture_present), y = mean_score, shape = groupstatus),
  position = position_dodge(width = 0.3),
  size = 3,
  color = "black",
  fill = "white"
) +

# 2d: Connecting lines between group means
geom_line(
  data = summary_data,
  aes(x = factor(rep_gesture_present), y = mean_score, group = groupstatus, linetype = groupstatus),
  position = position_dodge(width = 0.3),
  color = "black",
  size = 1
) +

# 2e: Labels and APA-style theme
scale_x_discrete(labels = c("No Gesture", "Gesture")) +

# IMPORTANT: match PL = dashed/triangle, TD = solid/circle
scale_shape_manual(values = c("PL" = 24, "TD" = 21)) + # triangle = 24, circle = 21
scale_linetype_manual(values = c("PL" = "dashed", "TD" = "solid")) +

labs(
  title = "Interaction of Group Status and Representational Gesture",
  x = "Representational Gesture Presence",
  y = "Narrative Structure Score",
  shape = "Group Status",
  linetype = "Group Status"
) +

theme_minimal(base_size = 14, base_family = "Times New Roman") +
theme(
  panel.background = element_rect(fill = "white", color = NA),
  plot.background = element_rect(fill = "white", color = NA),
  panel.grid.major = element_blank(),
  panel.grid.minor = element_blank(),
  panel.border = element_rect(color = "black", fill = NA),
  plot.title = element_text(hjust = 0.5, face = "bold"),
  axis.title = element_text(face = "bold"),
  axis.text = element_text(color = "black"),

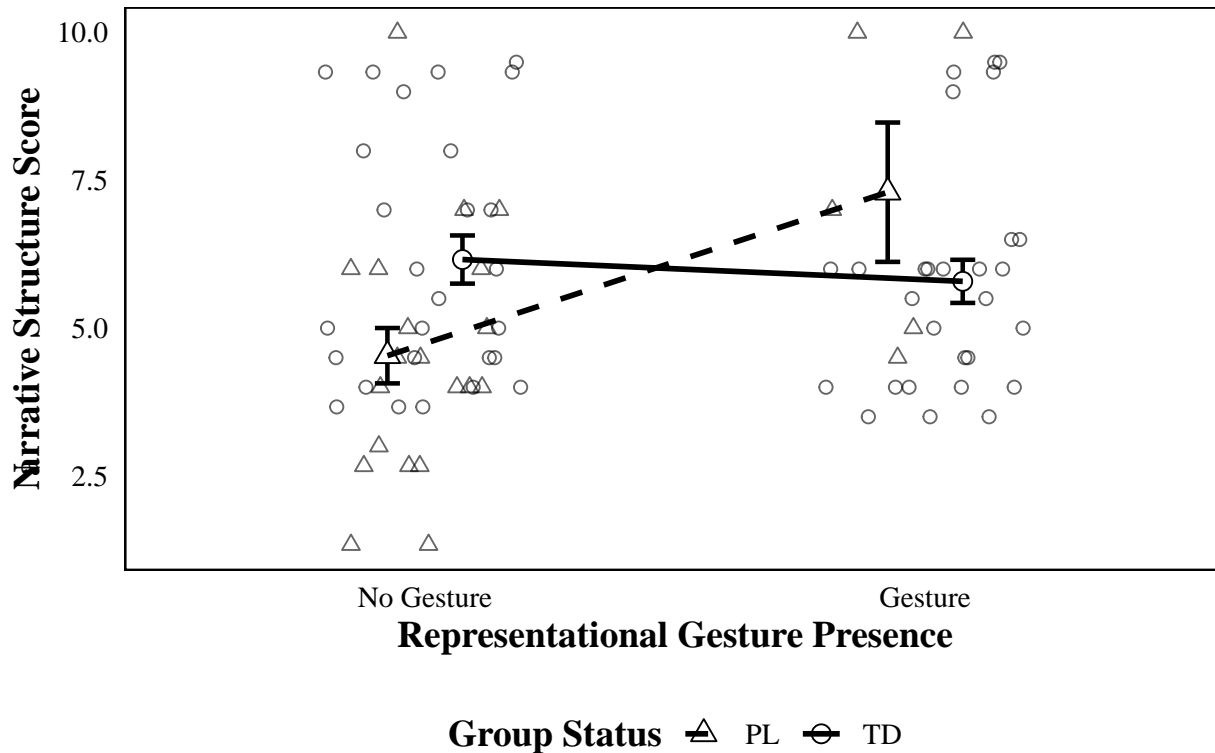
```

```

legend.position = "bottom",
legend.title = element_text(face = "bold")
)

```

Interaction of Group Status and Representational Gesture



```

# Figure X
#Observed mean narrative structure scores by group status and representational gesture presence during

##Figure X. Mean narrative structure scores by group and representational gesture presence. Solid and d

### Figure 2
# Interaction between group status and representational gesture presence predicting narrative structure

#ROBUST regressions ## Robust Regression for Model 1
library(robustbase)

##
## Attaching package: 'robustbase'
## The following object is masked from 'package:boot':
##
##     salinity
## The following object is masked from 'package:lmerTest':
##
##     carrots
modell1_robust <- lmrob(max_avg ~ groupstatus + total_gestures + total_pretend_episodes + mlu + groupsta
                        data = completely_merged_data_ALLyears)

```

```

# View summary
summary(model1_robust)

##
## Call:
## lmrob(formula = max_avg ~ groupstatus + total_gestures + total_pretend_episodes +
##       mlu + groupstatus:total_gestures, data = completely_merged_data_ALLyears)
## \--> method = "MM"
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.7047 -1.2849 -0.2128  1.3453  5.3023
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.2573698   0.5816078   5.601 3.45e-07 ***
## groupstatusTD    0.9805213   0.6821390   1.437  0.1548
## total_gestures    0.0303243   0.0287182   1.056  0.2944
## total_pretend_episodes 0.0002585   0.0030168   0.086  0.9319
## mlu              0.4512776   0.2017686   2.237  0.0283 *
## groupstatusTD:total_gestures -0.0446605   0.0273719  -1.632  0.1070
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Robust residual standard error: 1.858
## Multiple R-squared:  0.1391, Adjusted R-squared:  0.08095
## Convergence in 13 IRWLS iterations
##
## Robustness weights:
## 8 weights are ~= 1. The remaining 72 ones are summarized as
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.3957  0.8354  0.9512  0.8882  0.9753  0.9988
## Algorithmic parameters:
##      tuning.chi          bb      tuning.psi      refine.tol
##      1.548e+00      5.000e-01      4.685e+00      1.000e-07
##      rel.tol          scale.tol      solve.tol      zero.tol
##      1.000e-07      1.000e-10      1.000e-07      1.000e-10
##      eps.outlier          eps.x warn.limit.reject warn.limit.meanrw
##      1.250e-03      1.002e-09      5.000e-01      5.000e-01
##      nResample      max.it      best.r.s      k.fast.s      k.max
##      500          50          2          1          200
##      maxit.scale      trace.lev      mts      compute.rd fast.s.large.n
##      200          0          1000          0          2000
##      psi          subsampling          cov
##      "bisquare"      "nonsingular"      ".vcov.avar1"
## compute.outlier.stats
##      "SM"
## seed : int(0)

```

Table for Model 1 Robust Regression

```

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

```

```

# Step 1: Tidy and prepare the model1_robust
modell_robust_table <- tidy(model1_robust) %>%
  mutate(
    term = case_when(
      term == "(Intercept)" ~ "Intercept",
      term == "groupstatusPL" ~ "Group Status (PL vs TD)",
      term == "total_gestures" ~ "Total Gestures",
      term == "total_pretend_episodes" ~ "Total Pretend Episodes",
      term == "mlu" ~ "Mean Length of Utterance (MLU)",
      term == "groupstatusPL:total_gestures" ~ "Group × Total Gestures Interaction",
      TRUE ~ term
    ),
    estimate = sprintf("%.2f", estimate), # Format b values
    std.error = sprintf("%.2f", std.error), # Format SE
    statistic = sprintf("%.2f", statistic), # Format t
    p.value = as.numeric(p.value) # Keep numeric for sig flag
  ) %>%
  mutate(
    sig = ifelse(p.value < 0.05, TRUE, FALSE), # Flag significance
    p.value = ifelse(p.value < .001, "< .001", sprintf("%.3f", p.value)) # Format p values
  )

# Step 2: Build the gt table with hidden sig column
modell_robust_table %>%
  select(Predictor = term, b = estimate, SE = std.error, t = statistic, p = p.value, sig) %>%
  gt() %>%
  tab_header(
    title = "Table X",
    subtitle = "Robust Regression Predicting Narrative Structure Scores (Model 1: Total Gestures)"
  ) %>%
  cols_label(
    Predictor = "Predictor",
    b = "b",
    SE = "SE",
    t = "t",
    p = "p"
  ) %>%
  tab_style(
    style = cell_text(weight = "bold"),
    locations = cells_body(
      rows = sig == TRUE
    )
  ) %>%
  cols_hide(columns = sig) %>%
  tab_options(
    table.font.size = "small",
    heading.align = "center",
    data_row.padding = px(2),
    column_labels.font.weight = "bold",
    table.width = pct(80)
  ) %>%
  tab_source_note(
    source_note = "Note. b = unstandardized regression coefficient; SE = robust standard error; p = sig"
  )

```


Table X
Robust Regression Predicting Narrative Structure Scores (Model 1: Total Gestures)

Predictor	b	SE	t	p
Intercept	3.26	0.58	5.60	< .001
groupstatusTD	0.98	0.68	1.44	0.155
Total Gestures	0.03	0.03	1.06	0.294
Total Pretend Episodes	0.00	0.00	0.09	0.932
Mean Length of Utterance (MLU)	0.45	0.20	2.24	0.028
groupstatusTD:total_gestures	-0.04	0.03	-1.63	0.107

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

)

Robust Regression for Model 2

```
# Load package
library(robustbase)

# Run a robust linear model (rlm)
model2_robust <- lmrob(max_avg ~ groupstatus * rep_gesture_present + total_pretend_episodes + mlu,
                      data = completely_merged_data_ALLyears)

# View summary
summary(model2_robust)

##
## Call:
## lmrob(formula = max_avg ~ groupstatus * rep_gesture_present + total_pretend_episodes +
##       mlu, data = completely_merged_data_ALLyears)
## \--> method = "MM"
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.32315 -1.23990 -0.01447  1.26980  5.53153
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.3549590   0.5962652   5.627 3.11e-07 ***
## groupstatusTD    1.2008185   0.6761636   1.776  0.0799 .
## rep_gesture_present 2.6412380   1.9266191   1.371  0.1745
## total_pretend_episodes -0.0008886   0.0025567  -0.348  0.7292
## mlu             0.4165026   0.2108196   1.976  0.0519 .
## groupstatusTD:rep_gesture_present -3.0930558   1.8702551  -1.654  0.1024
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Robust residual standard error: 1.881
## Multiple R-squared:  0.1772, Adjusted R-squared:  0.1216
## Convergence in 17 IRWLS iterations
##
```

```
## Robustness weights:
## 8 weights are ~= 1. The remaining 72 ones are summarized as
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.3673 0.8215 0.9511 0.8953 0.9803 0.9986
## Algorithmic parameters:
##      tuning.chi          bb      tuning.psi      refine.tol
##      1.548e+00      5.000e-01      4.685e+00      1.000e-07
##      rel.tol      scale.tol      solve.tol      zero.tol
##      1.000e-07      1.000e-10      1.000e-07      1.000e-10
##      eps.outlier      eps.x warn.limit.reject warn.limit.meanrw
##      1.250e-03      1.002e-09      5.000e-01      5.000e-01
##      nResample      max.it      best.r.s      k.fast.s      k.max
##      500      50      2      1      200
##      maxit.scale      trace.lev      mts      compute.rd fast.s.large.n
##      200      0      1000      0      2000
##      psi      subsampling      cov
##      "bisquare"      "nonsingular"      ".vcov.avar1"
## compute.outlier.stats
##      "SM"
## seed : int(0)
```

Table for Model 2 Robust Regression

```
# Assuming the same `model2_table` as above
library(broom)
library(dplyr)
library(gt)

# Step 1: Tidy and prepare the model
model2_robust_table <- tidy(model2_robust) %>%
  mutate(
    term = case_when(
      term == "(Intercept)" ~ "Intercept",
      term == "groupstatusPL" ~ "Group Status (PL vs TD)",
      term == "rep_gesture_present" ~ "Representative Gesture",
      term == "total_pretend_episodes" ~ "Total Pretend Episodes",
      term == "mlu" ~ "Mean Length of Utterance (MLU)",
      term == "groupstatusPL:rep_gesture_present" ~ "Group × Representative Gesture Interaction",
      TRUE ~ term
    ),
    estimate = sprintf("%.2f", estimate),
    std.error = sprintf("%.2f", std.error),
    statistic = sprintf("%.2f", statistic),
    p.value = as.numeric(p.value) # keep numeric for now
  ) %>%
  mutate(
    sig = ifelse(p.value < 0.05, TRUE, FALSE), # Flag significance
    p.value = ifelse(p.value < .001, "< .001", sprintf("%.3f", p.value)) # Format p
  )

# Step 2: Build table - but select only the columns you want shown
model2_robust_table %>%
  select(Predictor = term, b = estimate, SE = std.error, t = statistic, p = p.value, sig) %>% # include
  gt() %>%
```

Table X
Robust Regression Predicting Narrative Structure Scores

Predictor	b	SE	t	p
Intercept	3.35	0.60	5.63	< .001
groupstatusTD	1.20	0.68	1.78	0.080
Representative Gesture	2.64	1.93	1.37	0.175
Total Pretend Episodes	-0.00	0.00	-0.35	0.729
Mean Length of Utterance (MLU)	0.42	0.21	1.98	0.052
groupstatusTD:rep_gesture_present	-3.09	1.87	-1.65	0.102

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

```

tab_header(
  title = "Table X",
  subtitle = "Robust 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(
    rows = sig == TRUE
  )
) %>%
cols_hide(columns = sig) %>% # <<< THIS hides the sig column from printing
tab_options(
  table.font.size = "small",
  heading.align = "center",
  data_row.padding = px(2),
  column_labels.font.weight = "bold",
  table.width = pct(80)
) %>%
tab_source_note(
  source_note = "Note. b = unstandardized regression coefficient; SE = robust standard error; p = significance level. Significant predictors are bolded."
)

```

Robust Regression for Reduced Model 2

```

# Robust regression for reduced Model 2 (your final one)
model2_reduced_robust <- lmrob(max_avg ~ groupstatus + rep_gesture_present + mlu + groupstatus:rep_gesture_present,
  data = completely_merged_data_ALLyears)
summary(model2_reduced_robust)

```

```
##
## Call:
```

```
## lmrob(formula = max_avg ~ groupstatus + rep_gesture_present + mlu + groupstatus:rep_gesture_present,
##       data = completely_merged_data_ALLyears)
## --> method = "MM"
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.34340 -1.25349 -0.05977  1.30482  5.48662
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      3.2929     0.5717   5.760 1.75e-07 ***
## groupstatusTD      1.1943     0.6773   1.763  0.0819 .
## rep_gesture_present  2.4475     1.5909   1.538  0.1282
## mlu                0.4175     0.2129   1.961  0.0535 .
## groupstatusTD:rep_gesture_present -2.9549     1.6895  -1.749  0.0844 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Robust residual standard error: 1.896
## Multiple R-squared:  0.1736, Adjusted R-squared:  0.1295
## Convergence in 15 IRWLS iterations
##
## Robustness weights:
## 9 weights are ~= 1. The remaining 71 ones are summarized as
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.3826  0.8276  0.9463  0.8952  0.9806  0.9987
## Algorithmic parameters:
##      tuning.chi          bb      tuning.psi      refine.tol
##      1.548e+00      5.000e-01      4.685e+00      1.000e-07
##      rel.tol      scale.tol      solve.tol      zero.tol
##      1.000e-07      1.000e-10      1.000e-07      1.000e-10
##      eps.outlier      eps.x warn.limit.reject warn.limit.meanrw
##      1.250e-03      1.169e-11      5.000e-01      5.000e-01
##      nResample      max.it      best.r.s      k.fast.s      k.max
##      500          50          2          1          200
##      maxit.scale      trace.lev      mts      compute.rd fast.s.large.n
##      200          0          1000      0          2000
##      psi      subsampling      cov
##      "bisquare"      "nonsingular"      ".vcov.avar1"
## compute.outlier.stats
##      "SM"
## seed : int(0)
```

Model 2 (reduced) Side by Side Regresssion Table of OLS and Robust

```
# Load libraries
library(broom)
library(dplyr)

# Tidy the original OLS model
ols_table <- tidy(model2_ols_lean_noyear) %>%
  mutate(
    Model = "OLS",
    Estimate = round(estimate, 2),
    SE = round(std.error, 2),
```

```

    p = ifelse(p.value < .001, "< .001", sprintf("%.3f", p.value))
  ) %>%
  select(Model, term, Estimate, SE, p)

# Tidy the robust model
robust_table <- tidy(model2_reduced_robust) %>%
  mutate(
    Model = "Robust",
    Estimate = round(estimate, 2),
    SE = round(std.error, 2),
    p = ifelse(p.value < .001, "< .001", sprintf("%.3f", p.value))
  ) %>%
  select(Model, term, Estimate, SE, p)

# Combine
combined_table <- bind_rows(ols_table, robust_table) %>%
  arrange(term, Model) # Keep terms together

# Load gt package
library(gt)

# Build the table
combined_table %>%
  rename(
    Predictor = term
  ) %>%
  gt(groupname_col = "Model") %>%
  tab_header(
    title = "Table X",
    subtitle = "Comparison of OLS and Robust Regression Estimates for Reduced Model 2"
  ) %>%
  cols_label(
    Predictor = "Predictor",
    Estimate = "b",
    SE = "SE",
    p = "p"
  ) %>%
  tab_options(
    table.font.size = "small",
    data_row.padding = px(2),
    table.align = "center"
  ) %>%
  tab_source_note(
    source_note = "Note. b = unstandardized regression coefficient; SE = standard error; p = significant"
  )

```

Models 1 & 2(og) Combined Table

```

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

# Step 1: Tidy and prep each model separately

```

Table X
Comparison of OLS and Robust Regression Estimates for Reduced Model 2

Predictor	b	SE	p
OLS			
(Intercept)	3.54	0.70	< .001
groupstatusTD	1.28	0.63	0.046
groupstatusTD:rep_gesture_present	-2.88	1.17	0.016
mlu	0.35	0.19	0.068
rep_gesture_present	2.43	1.04	0.022
Robust			
(Intercept)	3.29	0.57	< .001
groupstatusTD	1.19	0.68	0.082
groupstatusTD:rep_gesture_present	-2.95	1.69	0.084
mlu	0.42	0.21	0.054
rep_gesture_present	2.45	1.59	0.128

Note. b = unstandardized regression coefficient; SE = standard error; p = significance level. Robust regression conducted using MM-estimation with bisquare psi function.

```

modell1_robust_tidy <- tidy(modell1_robust) %>%
  mutate(
    Predictor = case_when(
      term == "(Intercept)" ~ "Intercept",
      term == "groupstatusPL" ~ "Group Status (PL vs TD)",
      term == "total_gestures" ~ "Total Gestures",
      term == "total_pretend_episodes" ~ "Total Pretend Episodes",
      term == "mlu" ~ "Mean Length of Utterance (MLU)",
      term == "groupstatusPL:total_gestures" ~ "Group × Total Gestures Interaction",
      TRUE ~ term
    ),
    Model = "Model 1"
  ) %>%
  select(Predictor, estimate, std.error, p.value)

modell2_robust_tidy <- tidy(modell2_robust) %>%
  mutate(
    Predictor = case_when(
      term == "(Intercept)" ~ "Intercept",
      term == "groupstatusPL" ~ "Group Status (PL vs TD)",
      term == "rep_gesture_present" ~ "Representative Gesture",
      term == "total_pretend_episodes" ~ "Total Pretend Episodes",
      term == "mlu" ~ "Mean Length of Utterance (MLU)",
      term == "groupstatusPL:rep_gesture_present" ~ "Group × Representative Gesture Interaction",
      TRUE ~ term
    ),
    Model = "Model 2"
  ) %>%
  select(Predictor, estimate, std.error, p.value)

# Step 2: Full join based on predictor names

```

```

combined_robust_table <- full_join(model1_robust_tidy, model2_robust_tidy, by = "Predictor", suffix = c
mutate(
  Sig1 = case_when(
    p.value_Model1 < .001 ~ "***",
    p.value_Model1 < .01 ~ "**",
    p.value_Model1 < .05 ~ "*",
    TRUE ~ ""
  ),
  Sig2 = case_when(
    p.value_Model2 < .001 ~ "***",
    p.value_Model2 < .01 ~ "**",
    p.value_Model2 < .05 ~ "*",
    TRUE ~ ""
  ),
  `b (Model 1)` = ifelse(!is.na(estimate_Model1), sprintf("%.2f", estimate_Model1), NA),
  `SE (Model 1)` = ifelse(!is.na(std.error_Model1), sprintf("%.2f", std.error_Model1), NA),
  `b (Model 2)` = ifelse(!is.na(estimate_Model2), sprintf("%.2f", estimate_Model2), NA),
  `SE (Model 2)` = ifelse(!is.na(std.error_Model2), sprintf("%.2f", std.error_Model2), NA),
  `b (Model 1)` = paste0(`b (Model 1)`, Sig1),
  `b (Model 2)` = paste0(`b (Model 2)`, Sig2)
) %>%
select(Predictor, `b (Model 1)`, `SE (Model 1)`, `b (Model 2)`, `SE (Model 2)`, p.value_Model1, p.val

# Step 3: Create the gt table
combined_robust_table %>%
gt() %>%
tab_header(
  title = "Table X",
  subtitle = "Robust Regression Predicting Narrative Structure Scores Across Models"
) %>%
cols_label(
  Predictor = "Predictor",
  `b (Model 1)` = "b (Model 1)",
  `SE (Model 1)` = "SE (Model 1)",
  `b (Model 2)` = "b (Model 2)",
  `SE (Model 2)` = "SE (Model 2)"
) %>%
tab_style(
  style = cell_text(weight = "bold"),
  locations = cells_body(
    rows = (p.value_Model1 < 0.05) | (p.value_Model2 < 0.05),
    columns = c(`b (Model 1)`, `b (Model 2)`)
  )
) %>%
cols_hide(columns = c(p.value_Model1, p.value_Model2)) %>%
tab_options(
  table.font.size = "small",
  heading.align = "center",
  data_row.padding = px(2),
  column_labels.font.weight = "bold",
  table.width = pct(90)
) %>%
tab_source_note(

```

Table X
Robust Regression Predicting Narrative Structure Scores Across Models

Predictor	b (Model 1)	SE (Model 1)	b (Model 2)	SE (Model 2)
Intercept	3.26***	0.58	3.35***	0.60
groupstatusTD	0.98	0.68	1.20	0.68
Total Gestures	0.03	0.03	NA	NA
Total Pretend Episodes	0.00	0.00	-0.00	0.00
Mean Length of Utterance (MLU)	0.45*	0.20	0.42	0.21
groupstatusTD:total_gestures	-0.04	0.03	NA	NA
Representative Gesture	NA	NA	2.64	1.93
groupstatusTD:rep_gesture_present	NA	NA	-3.09	1.87

Note. b = unstandardized regression coefficient; SE = robust standard error. *p* < .05, **p** < .01, ***p*** < .001.

```
source_note = "Note. b = unstandardized regression coefficient; SE = robust standard error. *p* < .05, **p** < .01, ***p*** < .001."
)
```

Effect sizes for Robust Models 1 & 2(reduced)

```
# For Model 1: Total Gestures
model1_reduced <- lmrob(max_avg ~ groupstatus + total_gestures + total_pretend_episodes + mlu,
                        data = completely_merged_data_ALLyears)

r2_full_model1 <- summary(model1_robust)$r.squared
r2_reduced_model1 <- summary(model1_reduced)$r.squared

f2_model1 <- (r2_full_model1 - r2_reduced_model1) / (1 - r2_full_model1)
print(paste("Cohen's f^2 for Model 1 (Total Gestures):", round(f2_model1, 3)))

## [1] "Cohen's f^2 for Model 1 (Total Gestures): 0.02"

# For Model 2: Gesture Presence
model2_reduced <- lmrob(max_avg ~ groupstatus + rep_gesture_present + total_pretend_episodes + mlu,
                        data = completely_merged_data_ALLyears)

r2_full_model2 <- summary(model2_robust)$r.squared
r2_reduced_model2 <- summary(model2_reduced)$r.squared

f2_model2 <- (r2_full_model2 - r2_reduced_model2) / (1 - r2_full_model2)
print(paste("Cohen's f^2 for Model 2 (Gesture Presence):", round(f2_model2, 3)))

## [1] "Cohen's f^2 for Model 2 (Gesture Presence): 0.07"
```

MLU scatterplot

```
# Load necessary libraries
library(ggplot2)
library(dplyr)

# Assume your data is called 'completely_merged_data_ALLyears'
# and it includes 'mlu', 'max_avg', and 'groupstatus'
```



```

# Make sure groupstatus is a factor if needed
completely_merged_data_ALLyears <- completely_merged_data_ALLyears %>%
  mutate(groupstatus = factor(groupstatus, levels = c("PL", "TD")))

# Create APA-style scatterplot
ggplot(completely_merged_data_ALLyears, aes(x = mlu, y = max_avg, group = groupstatus)) +

  # 1. Individual participant points
  geom_point(
    aes(shape = groupstatus),
    size = 3,
    color = "black",
    fill = "white"
  ) +

  # 2. Regression lines (separately for each group)
  geom_smooth(
    aes(linetype = groupstatus),
    method = "lm",
    se = TRUE,
    color = "black",
    size = 1
  ) +

  # 3. Manual scales for shapes and linetypes to match your other figures
  scale_shape_manual(values = c("PL" = 24, "TD" = 21)) + # PL = triangle open, TD = circle open
  scale_linetype_manual(values = c("PL" = "dashed", "TD" = "solid")) +

  # 4. Labels
  labs(
    title = "Relationship Between MLU and Narrative Structure Score by Group",
    x = "Mean Length of Utterance (MLU)",
    y = "Narrative Structure Score",
    shape = "Group Status",
    linetype = "Group Status"
  ) +

  # 5. APA-style theming
  theme_minimal(base_size = 14, base_family = "Times New Roman") +
  theme(
    panel.background = element_rect(fill = "white", color = NA),
    plot.background = element_rect(fill = "white", color = NA),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    panel.border = element_rect(color = "black", fill = NA),
    plot.title = element_text(hjust = 0.5, face = "bold"),
    axis.title = element_text(face = "bold"),
    axis.text = element_text(color = "black"),
    legend.position = "bottom",
    legend.title = element_text(face = "bold")
  )

## `geom_smooth()` using formula = 'y ~ x'

```

Relationship Between MLU and Narrative Structure Score by Group

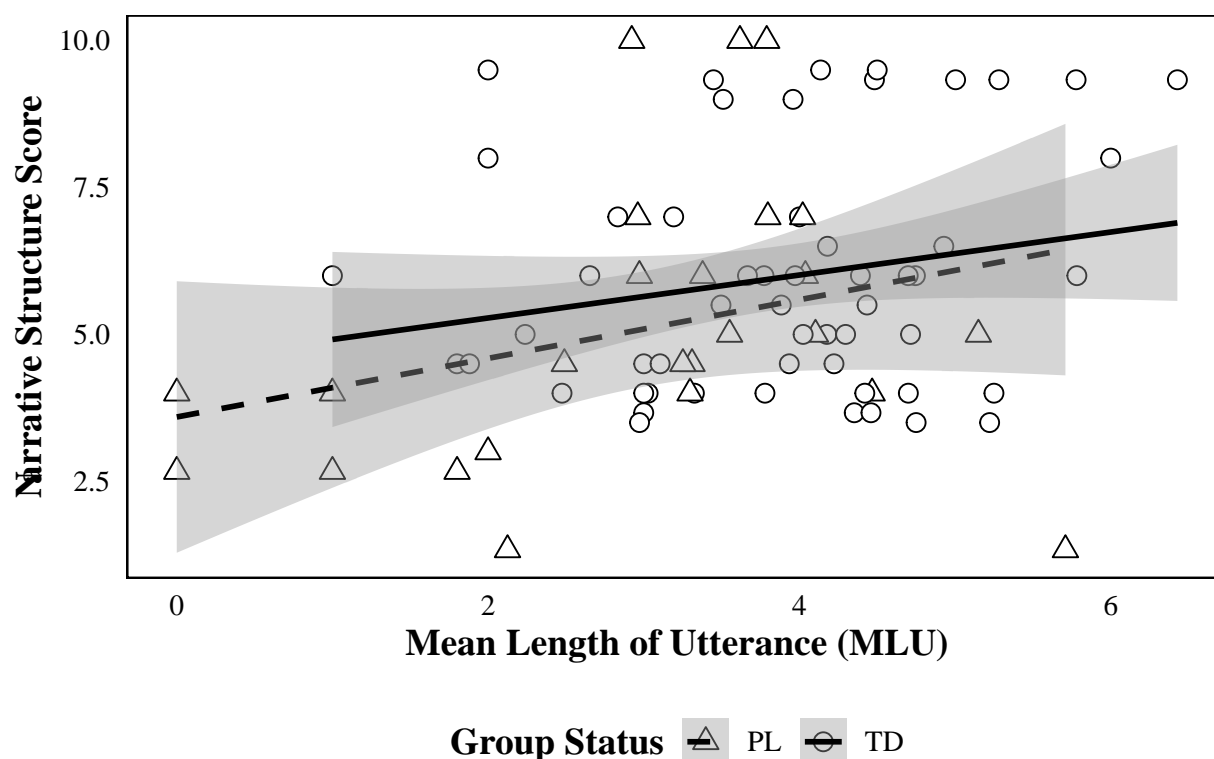


Figure 2

#Scatterplot of narrative structure scores as a function of mean length of utterance (MLU), separated by group status (PL and TD).

Text for Tables & Figures

Tables' Notes

For Table 1 (Main OLS Reduced Model):

Note. b = unstandardized regression coefficient; SE = standard error; p = significance level. Partial eta-squared effect sizes are reported in-text. $p < .05$. $\dagger p < .10$.

For Table A1 (OLS Model 1: Total Gestures):

Note. b = unstandardized regression coefficient; SE = standard error; p = significance level. No significant predictors emerged. $p < .05$. $\dagger p < .10$.

For Table A2 (Robust Model 1: Total Gestures):

Note. Robust regression estimated using M-estimation with bisquare weighting. No significant predictors emerged.

For Table A3 (Comparison of OLS and Robust for Reduced Model 2):

Note. Comparison of ordinary least squares (OLS) and robust regression estimates for reduced Model 2. Robust regression used M-estimation with bisquare weighting. b = unstandardized regression coefficient; SE = standard error; p = significance level. $p < .05$. $\dagger p < .10$.

Figures's Captions

Figure 1 (Observed Means Plot: Group × Gesture Presence)

Figure 1 Observed mean narrative structure scores by group status and representational gesture presence. Triangles and dashed lines represent children with perinatal lesions (PL); circles and solid lines represent typically developing (TD) children. Error bars represent ± 1 standard error of the mean.

Figure 2 (MLU Scatterplot)

Figure 2 Scatterplot of narrative structure scores as a function of mean length of utterance (MLU), separated by group status. Triangles and dashed lines represent children with perinatal lesions (PL); circles and solid lines represent typically developing (TD) children. Shaded regions represent 95% confidence intervals around fitted regression lines."

OLD CODE

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 = 0.3)) +
  stat_summary(fun.data = mean_se, geom = "errorbar", width = 0.1, position = position_dodge(width = 0.3)) +
  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)
```

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())
  )

# 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)
  )
```

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, linetype = groupstatus),
    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, color = groupstatus, linetype = groupstatus),
    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)
  )
)

```

fancy graph

```

# 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")
  )

```

fancy graph 2” Interaction of Group Status and Representational Gesture Presence”

```
# 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,
      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")
)

```

APA figure of Simple Slope: Representational Gestures vs. Narrative Score

```

# Step 1: Generate predicted values (simple slope for rep_gesture_present)
simple_slopes_rep_gesture <- ggpredict(
  model2,
  terms = c("rep_gesture_present", "groupstatus") # Gesture presence (0/1) x Group
)

# Step 2: Build APA-style figure
ggplot(simple_slopes_rep_gesture, aes(x = x, y = predicted, group = group)) +
  geom_ribbon(aes(ymin = conf.low, ymax = conf.high, fill = group),
    alpha = 0.1, color = NA) +
  geom_line(aes(linetype = group), color = "black", linewidth = 1.2) +
  geom_point(
    data = simple_slopes_rep_gesture %>%
      group_by(group) %>%
      filter(x == min(x) | x == max(x)), # Only at 0 (No Gesture) and 1 (Gesture Present)
    aes(shape = group),
    size = 3,
    fill = "black",
    color = "black"
  ) +
  scale_x_continuous(
    breaks = c(0, 1),
    labels = c("No Gesture", "Gesture")
  ) +
  scale_linetype_manual(
    values = c("solid", "dashed")
  ) +
  scale_shape_manual(
    values = c(21, 24) # circle and triangle
  ) +
  scale_fill_manual(
    values = c("gray70", "gray85")
  ) +
  labs(
    title = NULL,

```



```

x = "Representational Gesture Presence",
y = "Predicted Narrative Structure Score",
linetype = "Group Status",
shape = "Group Status",
fill = "Group Status"
) +
theme_minimal(base_size = 12) +
theme(
  panel.grid.major = element_blank(),
  panel.grid.minor = element_blank(),
  panel.background = element_rect(fill = "white", color = NA),
  plot.background = element_rect(fill = "white", color = NA),
  legend.position = "bottom",
  legend.title = element_text(face = "bold"),
  text = element_text(family = "Times New Roman"),
  axis.title = element_text(face = "bold"),
  axis.text = element_text(color = "black")
)

```

#Figure X. Predicted narrative structure scores by group and representational gesture presence. Solid a

APA simple slopes for total gestures & group status

```

# Step 1: Generate predictions across a smooth range
simple_slopes_total_gestures <- ggpredict(
  model1,
  terms = c("total_gestures [0:100 by=1]", "groupstatus")
)

# Step 2: Build APA-Perfect figure
ggplot(simple_slopes_total_gestures, aes(x = x, y = predicted, group = group)) +
  geom_ribbon(aes(ymin = conf.low, ymax = conf.high, fill = group),
    alpha = 0.1, color = NA) + # Light ribbons per group
  geom_line(aes(linetype = group), color = "black", linewidth = 1.2) + # black solid/dashed
  geom_point(
    data = simple_slopes_total_gestures %>%
      group_by(group) %>%
      filter(x == min(x) | x == max(x)), # Only mark start and end points
    aes(shape = group),
    size = 3,
    fill = "black",
    color = "black"
  ) +
  scale_linetype_manual(
    values = c("solid", "dashed") # PL = solid, TD = dashed
  ) +
  scale_shape_manual(
    values = c(21, 24) # Circle for one group, triangle for the other
  ) +
  scale_fill_manual(
    values = c("gray70", "gray85")
  ) +
  labs(

```

```

title = NULL,
x = "Total Gestures During Pretend Play",
y = "Predicted Narrative Structure Score",
linetype = "Group Status",
shape = "Group Status",
fill = "Group Status"
) +
theme_minimal(base_size = 12) +
theme(
  panel.grid.major = element_blank(),
  panel.grid.minor = element_blank(),
  panel.background = element_rect(fill = "white", color = NA),
  plot.background = element_rect(fill = "white", color = NA),
  legend.position = "bottom",
  legend.title = element_text(face = "bold"),
  text = element_text(family = "Times New Roman"),
  axis.title = element_text(face = "bold"),
  axis.text = element_text(color = "black")
)

```

#Figure X. Predicted narrative structure scores by group and total number of gestures during pretend pl

figure of Simple Slope: Representational Gestures vs. Narrative Score

```

# Create sequence for representational gestures
rep_range <- seq(
  min(completely_merged_data_ALLyears$rep_gestures, na.rm = TRUE),
  max(completely_merged_data_ALLyears$rep_gestures, na.rm = TRUE),
  length.out = 100
)

# Build new prediction dataset
pred_data_rep <- data.frame(
  groupstatus = factor("PL", levels = c("PL", "TD")), # match model factor
  total_gestures = mean(completely_merged_data_ALLyears$total_gestures, na.rm = TRUE), # control for t
  rep_gestures = rep_range, # vary rep gestures across observed range
  mlu = mean(completely_merged_data_ALLyears$mlu, na.rm = TRUE), # control for MLU
  total_pretend_episodes = mean(completely_merged_data_ALLyears$total_pretend_episodes, na.rm = TRUE),
  Year = factor("7", levels = c("7", "8", "10")) # hold year constant
)

# Predict narrative scores
pred_data_rep$predicted_narrative <- predict(model_fixed_effects, newdata = pred_data_rep)

# Plot
ggplot(completely_merged_data_ALLyears, aes(x = rep_gestures, y = max_avg)) +

  # Raw participant points
  geom_point(alpha = 0.6, color = "black", size = 2) +

  # Predicted regression line
  geom_line(data = pred_data_rep, aes(x = rep_gestures, y = predicted_narrative),
    color = "black", size = 1.2) +

```

```

# Labels
labs(
  title = "More representational gestures predict better narrative structure",
  x = "Number of representational gestures",
  y = "Narrative structure score"
) +

# APA-style theme
theme_minimal(base_size = 14, base_family = "Times New Roman") +
theme(
  panel.grid.major = element_blank(),
  panel.grid.minor = element_blank(),
  panel.border = element_rect(color = "black", fill = NA, size = 0.8),
  axis.line = element_line(color = "black"),
  plot.title = element_text(hjust = 0.5, face = "bold"),
  axis.title = element_text(face = "bold"),
  legend.position = "none"
)

```

```

library(ggplot2)

ggplot(completely_merged_data_ALLyears, aes(x = rep_gestures, y = max_avg)) +

# Raw participant points (still black dots)
geom_point(alpha = 0.6, color = "black", size = 2) +

# Participant IDs as text labels slightly above the dots
geom_text(aes(label = participant_id),
  vjust = -1, # vertical adjustment above the point
  size = 3, # size of text (adjustable)
  family = "Times New Roman") +

# Predicted regression line
geom_line(data = pred_data_rep, aes(x = rep_gestures, y = predicted_narrative),
  color = "black", size = 1.2) +

# Labels
labs(
  title = "More representational gestures predict better narrative structure",
  x = "Number of representational gestures",
  y = "Narrative structure score"
) +

# APA-style minimal theme
theme_minimal(base_size = 14, base_family = "Times New Roman") +
theme(
  panel.grid.major = element_blank(),
  panel.grid.minor = element_blank(),
  panel.border = element_rect(color = "black", fill = NA, size = 0.8),
  axis.line = element_line(color = "black"),
  plot.title = element_text(hjust = 0.5, face = "bold"),
  axis.title = element_text(face = "bold"),
  legend.position = "none"
)

```

```
)
```

simple slope of Total Gestures vs. Narrative Score

```
# Create a sequence of total gestures (z-scored)
gestures_range <- seq(
  min(completely_merged_data_ALLyears$total_gestures, na.rm = TRUE),
  max(completely_merged_data_ALLyears$total_gestures, na.rm = TRUE),
  length.out = 100
)

# Build prediction data
pred_data_total <- data.frame(
  groupstatus = factor("PL", levels = c("PL", "TD")), # Hold group constant
  total_gestures = gestures_range, # Vary total gestures
  rep_gestures = mean(completely_merged_data_ALLyears$rep_gestures, na.rm = TRUE), # Hold rep gestures
  mlu_z = mean(completely_merged_data_ALLyears$mlu_z, na.rm = TRUE), # Hold MLU constant
  total_pretend_episodes = mean(completely_merged_data_ALLyears$total_pretend_episodes, na.rm = TRUE),
  Year = factor("7", levels = c("7", "8", "10")) # Hold year constant
)

# Predict narrative scores
pred_data_total$predicted_narrative <- predict(model_fixed_effects, newdata = pred_data_total)

ggplot(completely_merged_data_ALLyears, aes(x = total_gestures, y = max_avg)) +

  # Raw participant points
  geom_point(alpha = 0.6, color = "black", size = 2) +

  # Predicted regression line
  geom_line(data = pred_data_total, aes(x = total_gestures, y = predicted_narrative),
    color = "black", size = 1.2) +

  # Labels
  labs(
    title = "Total gestures show marginal negative trend with narrative structure",
    x = "Total gestures during pretend play (z-scored)",
    y = "Narrative structure score"
  ) +

  # APA-style minimal theme
  theme_minimal(base_size = 14, base_family = "Times New Roman") +
  theme(
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    panel.border = element_rect(color = "black", fill = NA, size = 0.8),
    axis.line = element_line(color = "black"),
    plot.title = element_text(hjust = 0.5, face = "bold"),
    axis.title = element_text(face = "bold"),
    legend.position = "none"
  )
)
```

Conceptual diagram of moderation model:

```
library(DiagrammerR)

grViz("
digraph moderation_model {

  # General node style
  node [shape = rectangle, fontname = 'Times New Roman', fontsize = 14]

  # Nodes
  GesturePresence [label = 'Representational Gesture Presence']
  GroupStatus [label = 'Group Status (PL vs TD)']
  NarrativePL [label = 'Narrative Structure Score (PL)']
  NarrativeTD [label = 'Narrative Structure Score (TD)']

  # Solid Arrows (Main Effects)
  GesturePresence -> NarrativePL [label = 'Stronger Positive Effect', style = solid]
  GesturePresence -> NarrativeTD [label = 'Weaker Positive Effect', style = solid]

  # Dashed Arrows (Moderation)
  GroupStatus -> NarrativePL [label = 'Moderates', style = dashed]
  GroupStatus -> NarrativeTD [label = 'Moderates', style = dashed]
}
")

#Figure 3
#Conceptual model illustrating the moderating effect of group status (PL vs. TD) on the relationship be
```

BOOTSTRAPPING!!! (...won't work)

```
library(boot)          # For bootstrapping
library(interactions)  # For simple slopes and plotting

simple_slope_function <- function(data, indices) {
  d <- data[indices, ] # Resample data
  tryCatch({
    # Refit YOUR model structure
    fit <- lm(max_avg ~ groupstatus + rep_gesture_present + total_pretend_episodes + mlu + groupstatus:rep_gesture_present,
              data = d)

    # Extract coefficients
    coefs <- coef(fit)

    # Simple slope for PL (baseline group)
    slope_PL <- coefs["rep_gesture_present"]

    # Simple slope for TD (PL slope + interaction)
    slope_TD <- coefs["rep_gesture_present"] + coefs["groupstatusTD:rep_gesture_present"]

    return(c(slope_PL, slope_TD))
  }, error = function(e) {
    # If the model fails to fit, return NA
  })
}
```

```

    return(c(NA, NA))
  })
}

set.seed(1234)

boot_results <- boot(
  data = completely_merged_data_ALLyears,
  statistic = simple_slope_function,
  R = 1000 # 1000 resamples
)

any(is.na(boot_results$t)) # TRUE if any NA slopes

# Clean if needed:
boot_clean <- boot_results
boot_clean$t <- boot_clean$t[complete.cases(boot_clean$t), ]

# For PL group (baseline)
boot.ci(boot_clean, type = "perc", index = 1)

# For TD group (interaction-adjusted)
boot.ci(boot_clean, type = "perc", index = 2)

```

(FAILED) 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 + group
  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,
  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_episodes_with_gesture_z_z", "mlu_z")]

```

```

# 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)

```