PretendPlay_Gesture_Analysis_ALLyears

Kristen Johnson

2025-04-25

SETUP

load 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
##
##
##
## Attaching package: 'rstatix'
##
##
## The following objects are masked from 'package:effectsize':
##
##     cohens_d, eta_squared
##
##
## The following object is masked from 'package:stats':
##
##     filter
```

```
##
##
## here() starts at /Users/kristenjohnson/KristenWorkingDirectory/Play_Narrative
##
## Registering fonts with R
```

Read in each dataset and add Year column

```
child_summary_H10 <- read_csv(here("PN_Datasets", "Pretend_Play", "CSVs_of_Combined_Data_PP", "child_su
```

```
## Rows: 28 Columns: 10
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): groupstatus
## dbl (8): participant_id, total_gestures, rep_gestures, total_pretend_episode...
## 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.
```

```
child_summary_H10$Year <- 10

child_summary_H8 <- read_csv(here("PN_Datasets", "Pretend_Play", "CSVs_of_Combined_Data_PP", "child_sum
```

```
## Rows: 23 Columns: 10
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): groupstatus
## dbl (8): participant_id, total_gestures, rep_gestures, total_pretend_episode...
## 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.
```

```
child_summary_H8$Year <- 8

child_summary_H7 <- read_csv(here("PN_Datasets", "Pretend_Play", "CSVs_of_Combined_Data_PP", "child_sum
```

```
## Rows: 29 Columns: 10
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): groupstatus
## dbl (8): participant_id, total_gestures, rep_gestures, total_pretend_episode...
## 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.
```

```
child_summary_H7$Year <- 7
```

Combine data across ALL THREE YEARS

```
# Combine all datasets into one data frame
pp_gesture_data_ALLyears <- bind_rows(child_summary_H10, child_summary_H8, child_summary_H7)
```

```r
# View the combined dataset
head(pp_gesture_data_ALLyears)
```

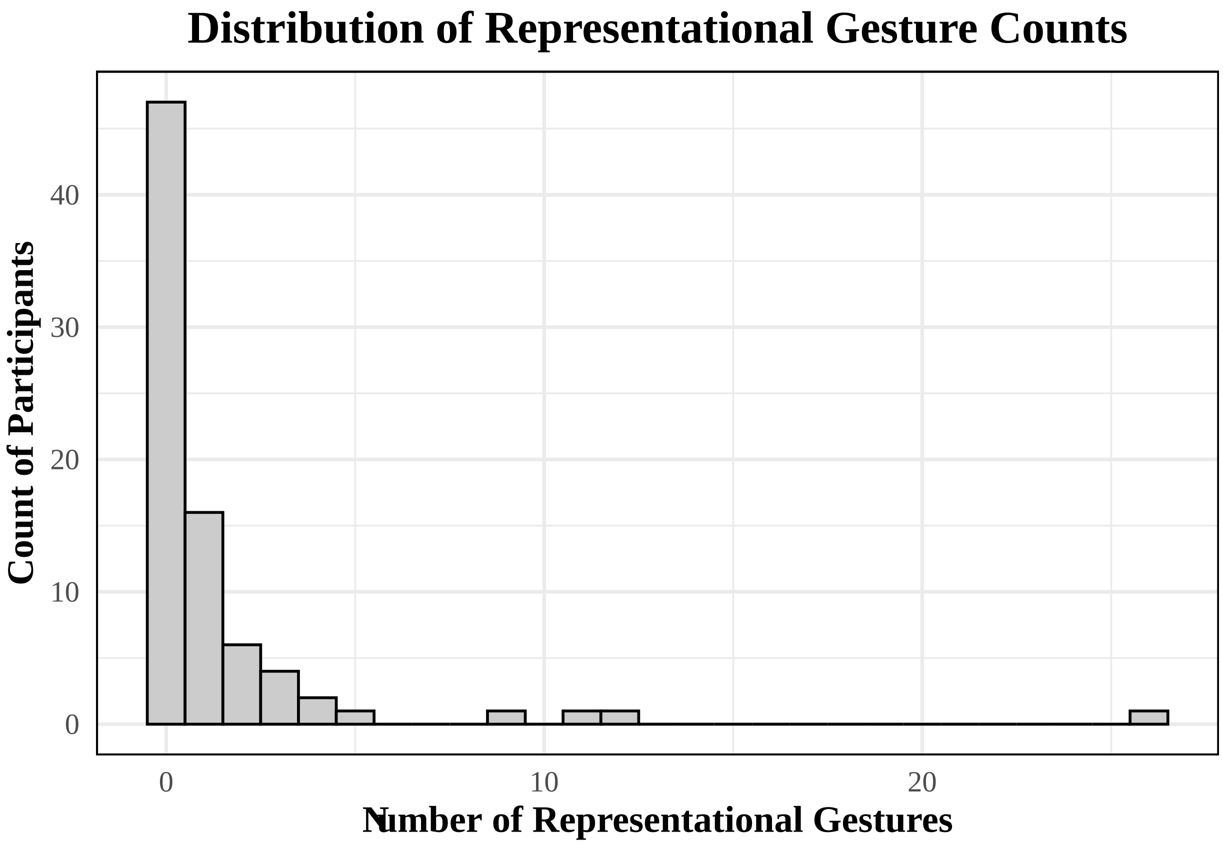
```
## # A tibble: 6 x 11
##   participant_id groupstatus total_gestures rep_gestures total_pretend_episodes
##            <dbl> <chr>                <dbl>        <dbl>                  <dbl>
## 1             22 TD                      12            1                    112
## 2             25 TD                       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 6 more variables: episodes_with_gesture <dbl>,
## #   prop_episodes_with_gesture <dbl>, prop_rep <dbl>, mlu <dbl>,
## #   mlu_missing_flag <lgl>, Year <dbl>
```

Check for Skewness

```r
 #Calculate skewness
skewness(pp_gesture_data_ALLyears$rep_gestures, na.rm = TRUE)
```

```
## [1] 4.82474
```

```r
# Histogram of rep_gestures
ggplot(pp_gesture_data_ALLyears, aes(x = rep_gestures)) +
  geom_histogram(binwidth = 1, fill = "gray80", color = "black") +
  labs(
    title = "Distribution of Representational Gesture Counts",
    x = "Number of Representational Gestures",
    y = "Count of Participants"
  ) +
  theme_minimal(base_size = 14, base_family = "Times New Roman") +
  theme(
    panel.border = element_rect(color = "black", fill = NA),
    plot.title = element_text(face = "bold", hjust = 0.5),
    axis.title = element_text(face = "bold")
  )
```



Correlations in Skewed Data ## Mann-Whitney U test on Rep Gestures (instead of ANOVA, bc highly skewed data)

```
# Step 1: Run the Mann-Whitney U test properly
mw_result <- wilcox.test(rep_gestures ~ groupstatus,
                         data = pp_gesture_data_ALLyears,
                         exact = FALSE)

# View the output if you want
print(mw_result)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  rep_gestures by groupstatus
## W = 508, p-value = 0.03643
## alternative hypothesis: true location shift is not equal to 0
```

```
# Step 2: Calculate total N
n1 <- pp_gesture_data_ALLyears %>%
  filter(groupstatus == "TD") %>%
  nrow()

n2 <- pp_gesture_data_ALLyears %>%
  filter(groupstatus == "PL") %>%
  nrow()

N <- n1 + n2
```

```r
# Step 3: Approximate Z from p-value
# p.value comes from mw_result
z_approx <- qnorm(mw_result$p.value / 2, lower.tail = FALSE)

# Step 4: Calculate r
r_effect_size <- z_approx / sqrt(N)

# Step 5: Print result
cat("Mann-Whitney U result: W =", mw_result$statistic,
    ", p =", signif(mw_result$p.value, 3),
    ", r =", round(r_effect_size, 2), "\n")
```

```
## Mann-Whitney U result: W = 508 , p = 0.0364 , r = 0.23
```

```r
# get medians for reporting (due to skewness)
pp_gesture_data_ALLyears %>%
  group_by(groupstatus) %>%
  summarise(Median_RepGest = median(rep_gestures, na.rm = TRUE))
```

```
## # A tibble: 2 x 2
##   groupstatus Median_RepGest
##   <chr>                <dbl>
## 1 PL                       0
## 2 TD                       1
```

Mann-Whitney U test on Total Gestures

```r
# Step 1: Run the Mann-Whitney U test properly
mw_result <- wilcox.test(total_gestures ~ groupstatus,
                         data = pp_gesture_data_ALLyears,
                         exact = FALSE)

# View the output if you want
print(mw_result)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  total_gestures by groupstatus
## W = 573, p-value = 0.2346
## alternative hypothesis: true location shift is not equal to 0
```

```r
# Step 2: Calculate total N
n1 <- pp_gesture_data_ALLyears %>%
  filter(groupstatus == "TD") %>%
  nrow()

n2 <- pp_gesture_data_ALLyears %>%
  filter(groupstatus == "PL") %>%
  nrow()

N <- n1 + n2

# Step 3: Approximate Z from p-value
# p.value comes from mw_result
```

```r
z_approx <- qnorm(mw_result$p.value / 2, lower.tail = FALSE)

# Step 4: Calculate r
r_effect_size <- z_approx / sqrt(N)

# Step 5: Print result
cat("Mann-Whitney U result: W =", mw_result$statistic,
    ", p =", signif(mw_result$p.value, 3),
    ", r =", round(r_effect_size, 2), "\n")
```

```
## Mann-Whitney U result: W = 573 , p = 0.235 , r = 0.13
```

```r
# get medians for reporting (due to skewness)
pp_gesture_data_ALLyears %>%
  group_by(groupstatus) %>%
  summarise(Median_TotGest = median(total_gestures, na.rm = TRUE))
```

```
## # A tibble: 2 x 2
##   groupstatus Median_TotGest
##   <chr>                <dbl>
## 1 PL                       3
## 2 TD                       6
```

Mann-Whitney U test on Proportion of Rep to Total Gesture

```r
# Mann-Whitney U for proportion of representative gestures
wilcox.test(prop_rep ~ groupstatus,
            data = pp_gesture_data_ALLyears,
            exact = FALSE)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  prop_rep by groupstatus
## W = 277.5, p-value = 0.01072
## alternative hypothesis: true location shift is not equal to 0
```

```r
# Calculate N
n1 <- pp_gesture_data_ALLyears %>% filter(groupstatus == "TD") %>% nrow()
n2 <- pp_gesture_data_ALLyears %>% filter(groupstatus == "PL") %>% nrow()
N <- n1 + n2

# Run Mann-Whitney test
mw_result_prop <- wilcox.test(prop_rep ~ groupstatus,
                              data = pp_gesture_data_ALLyears,
                              exact = FALSE)

# Approximate Z from p-value
z_approx_prop <- qnorm(mw_result_prop$p.value / 2, lower.tail = FALSE)

# Calculate r
r_effect_size_prop <- z_approx_prop / sqrt(N)

# Print results
cat("Mann-Whitney U result for proportion: W =", mw_result_prop$statistic,
    ", p =", signif(mw_result_prop$p.value, 3),
    ", r =", round(r_effect_size_prop, 2), "\n")
```

```
## Mann-Whitney U result for proportion: W = 277.5 , p = 0.0107 , r = 0.29
```

```
# get medians for reporting (due to skewness)
pp_gesture_data_ALLyears %>%
  group_by(groupstatus) %>%
  summarise(Median_PropRep = median(prop_rep, na.rm = TRUE))
```

```
## # A tibble: 2 x 2
##   groupstatus Median_PropRep
##   <chr>                <dbl>
## 1 PL                  0
## 2 TD                  0.0833
```

APA Table of ALL Results

```
# Load packages
library(dplyr)
library(gt)

# Create tidy data frame
gesture_results <- tibble::tibble(
  Variable = c("Total Gestures", "Representative Gestures", "Proportion of Representative Gestures"),
  PL_Median = c(3, 0, 0.00),
  TD_Median = c(6, 1, 0.08),
  W = c(573, 508, 277.5),
  p = c(0.235, 0.036, 0.011),
  r = c(0.13, 0.23, 0.29)
) %>%
  mutate(
    sig = ifelse(p < 0.05, TRUE, FALSE),
    p_formatted = ifelse(p < .001, "< .001", sprintf("%.3f", p)),
    r_formatted = sprintf("%.2f", r)
  )

# Build APA-style table: only bold p-values
gesture_results %>%
  select(Variable, PL_Median, TD_Median, W, p_formatted, r_formatted, sig) %>%
  rename(
    `Variable` = Variable,
    `Median (PL)` = PL_Median,
    `Median (TD)` = TD_Median,
    `W` = W,
    `p` = p_formatted,
    `r` = r_formatted
  ) %>%
  gt() %>%
  tab_header(
    title = md("**Table X**"),
    subtitle = md("_Mann-Whitney U Tests Comparing Gesture Measures Between Groups_")
  ) %>%
  tab_style(
```

Table X

Mann-Whitney U Tests Comparing Gesture Measures Between Groups

Variable	Median (PL)	Median (TD)	W	p	r
Total Gestures	3.00	6.00	573.00	0.235	0.13
Representative Gestures	0.00	1.00	508.00	**0.036**	0.23
Proportion of Representative Gestures	0.00	0.08	277.50	**0.011**	0.29

Note. *W* = Wilcoxon rank sum test statistic. *r* = effect size (rank biserial correlation). Bolded p-values indicate p < .05.

```r
    style = cell_text(font = "Times New Roman"),
    locations = cells_body()
  ) %>%
  tab_style(
    style = cell_text(weight = "bold"),
    locations = cells_body(
      columns = p,  # Bold only p-values
      rows = sig == TRUE
    )
  ) %>%
  tab_source_note(
    source_note = md("**Note.** _W_ = Wilcoxon rank sum test statistic. _r_ = effect size (rank biserial
  ) %>%
  fmt_number(
    columns = c(`Median (PL)`, `Median (TD)`, `W`, `r`),
    decimals = 2
  ) %>%
  cols_align(
    align = "left", columns = "Variable"
  ) %>%
  cols_align(
    align = "center", columns = c(`Median (PL)`, `Median (TD)`, `W`, `p`, `r`)
  ) %>%
  tab_options(
    table.font.size = "small",
    data_row.padding = px(2),
    table.align = "center",
    table.font.names = "Times New Roman"
  ) %>%
  cols_hide(columns = c(sig))  # Hide significance helper column
```

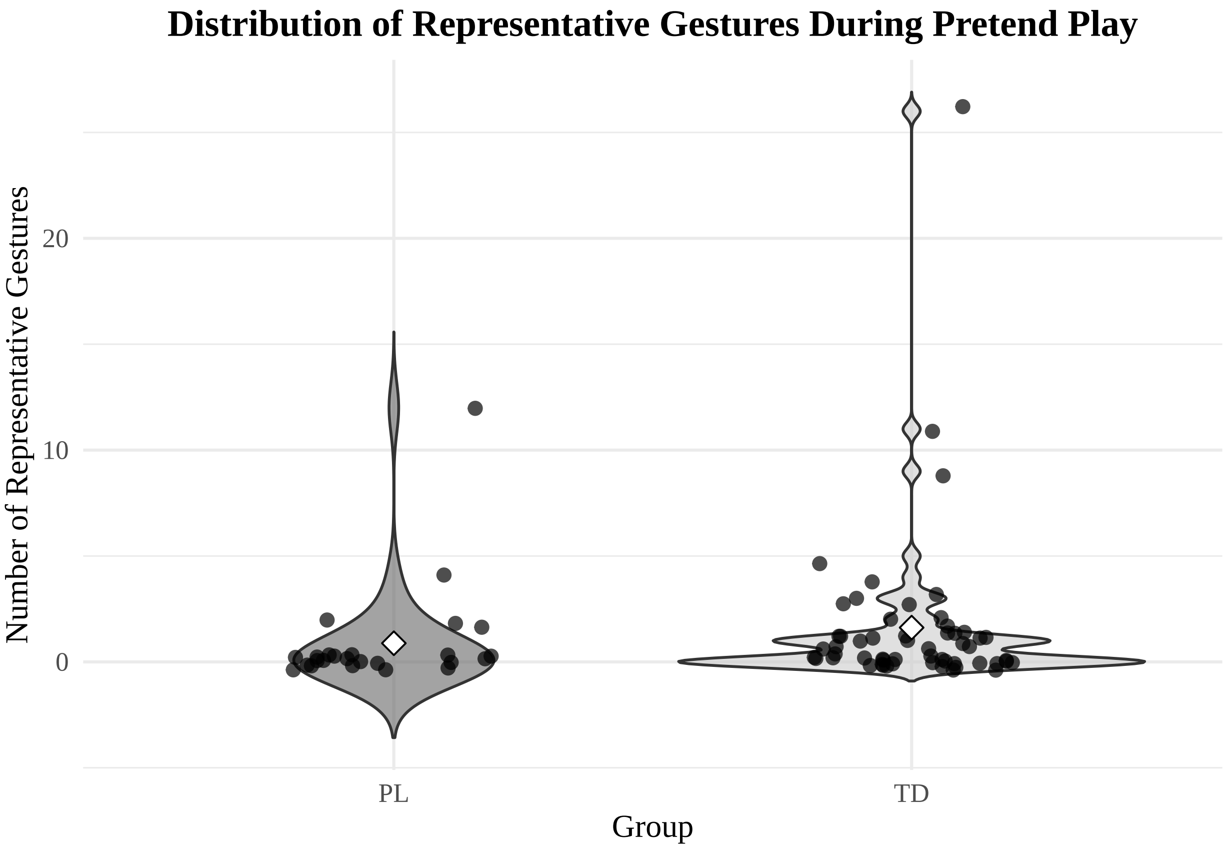
Means & SD's for ALL

```r
# Load dplyr
library(dplyr)

# Assuming your gesture_results or full dataset is called pp_gesture_data_ALLyears or similar
# Here's the simple code:

# Calculate means and SDs by group
gesture_summary_stats <- pp_gesture_data_ALLyears %>%
  group_by(groupstatus) %>%
  summarize(
    Mean_Total_Gestures = mean(total_gestures, na.rm = TRUE),
    SD_Total_Gestures = sd(total_gestures, na.rm = TRUE),
    Mean_Rep_Gestures = mean(rep_gestures, na.rm = TRUE),
    SD_Rep_Gestures = sd(rep_gestures, na.rm = TRUE),
    Mean_Prop_Rep = mean(prop_rep, na.rm = TRUE),
    SD_Prop_Rep = sd(prop_rep, na.rm = TRUE),
    n = n()
  )

# View the summary
gesture_summary_stats
```

```
## # A tibble: 2 x 8
##   groupstatus Mean_Total_Gestures SD_Total_Gestures Mean_Rep_Gestures
##   <chr>                     <dbl>             <dbl>             <dbl>
## 1 PL                         9.32              14.9              0.88
## 2 TD                        10.9               16.0              1.62
## # i 4 more variables: SD_Rep_Gestures <dbl>, Mean_Prop_Rep <dbl>,
## #   SD_Prop_Rep <dbl>, n <int>
```

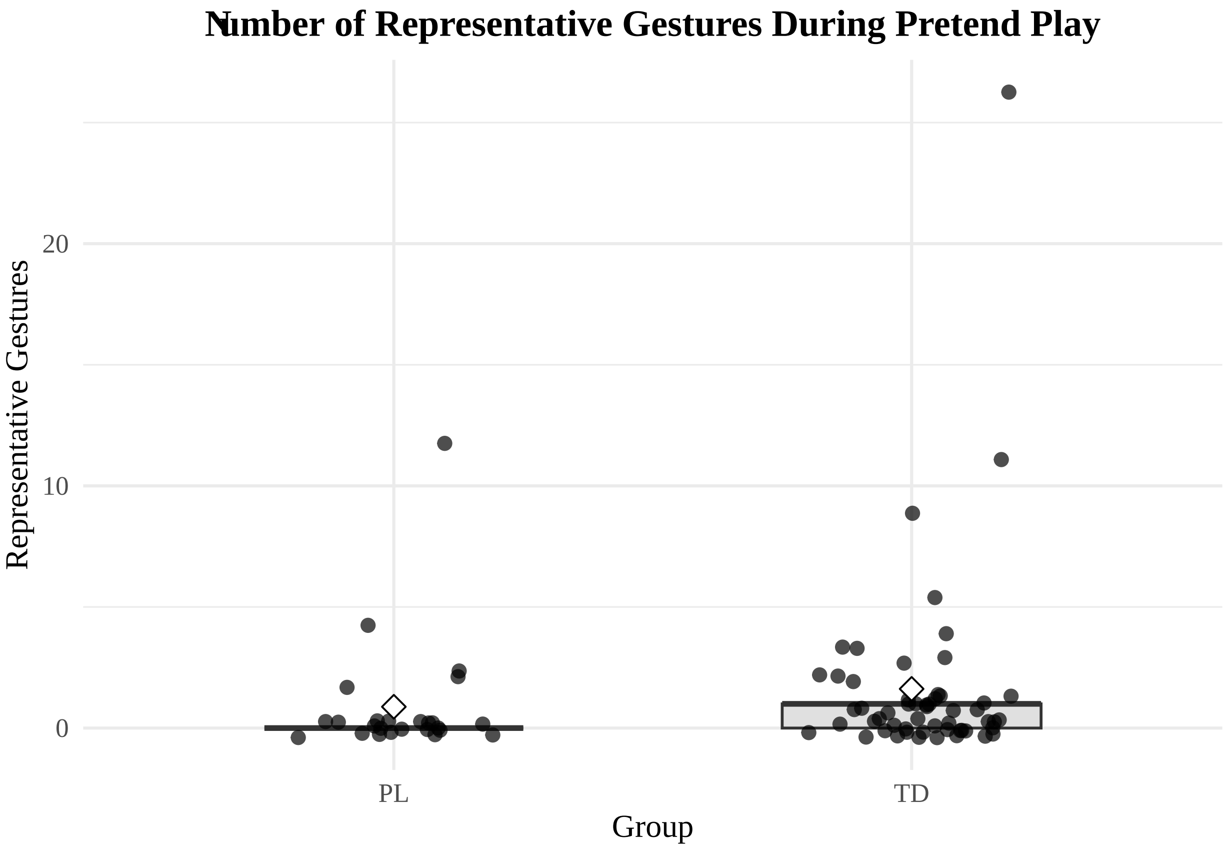
Violin Plot of Representative Gestures X Group Status

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

# Simple violin + jitter plot of representative gestures by group
pp_gesture_data_ALLyears %>%
  ggplot(aes(x = groupstatus, y = rep_gestures, fill = groupstatus)) +
  geom_violin(trim = FALSE, alpha = 0.6) +     # Violin for distribution
  geom_jitter(width = 0.2, size = 2, alpha = 0.7) +  # Individual points
  stat_summary(fun = mean, geom = "point", shape = 23, size = 3, fill = "white", color = "black") +  #
  labs(
    title = "Distribution of Representative Gestures During Pretend Play",
    x = "Group",
    y = "Number of Representative Gestures"
  ) +
  scale_fill_grey(start = 0.4, end = 0.8) +  # Subtle grayscale if you want APA look
  theme_minimal(base_size = 12) +
  theme(
    legend.position = "none",
    text = element_text(family = "Times New Roman"),
    plot.title = element_text(hjust = 0.5, face = "bold")
  )
```



Box Plot of Representative Gestures X Group Status

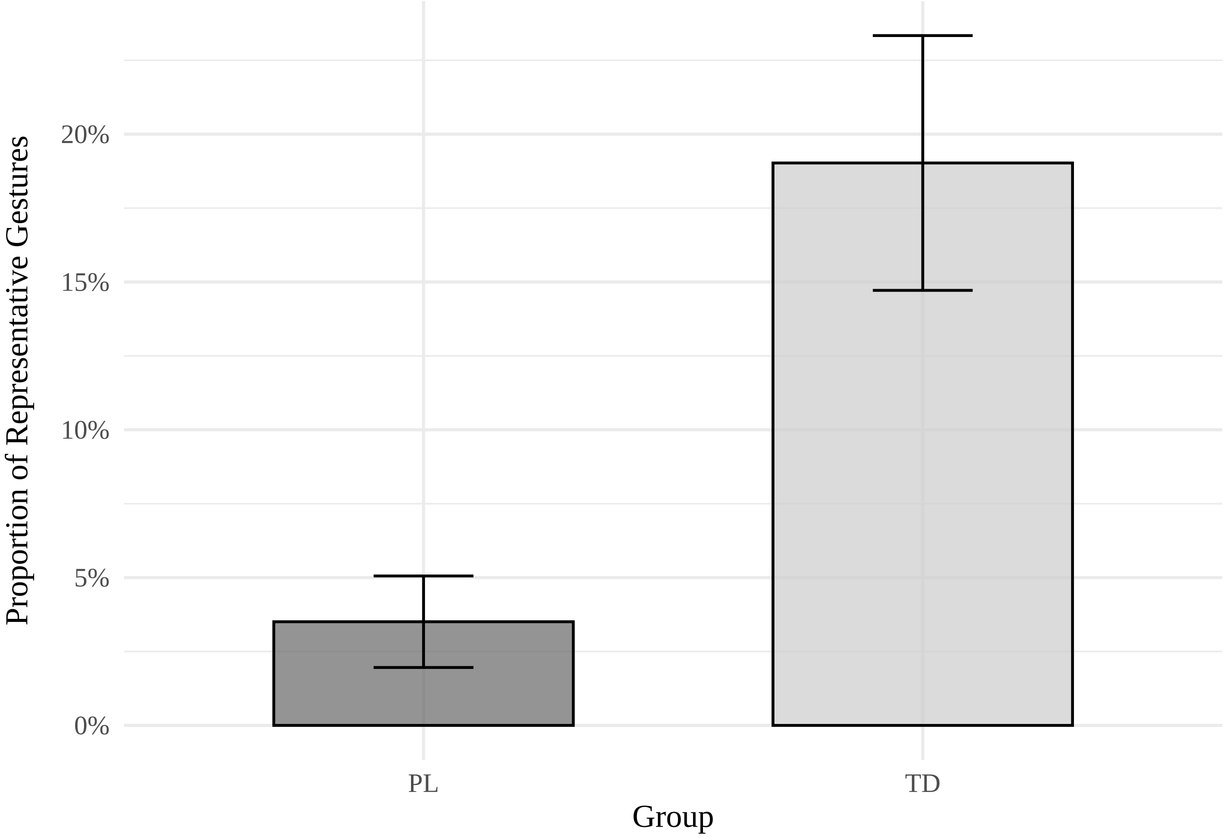
```
# APA-style boxplot of representative gestures by group
pp_gesture_data_ALLyears %>%
  ggplot(aes(x = groupstatus, y = rep_gestures, fill = groupstatus)) +
  geom_boxplot(width = 0.5, outlier.shape = NA, alpha = 0.6) +  # No outlier points (APA-style boxplots
  geom_jitter(width = 0.2, size = 2, alpha = 0.7) +  # Add individual points
  stat_summary(fun = mean, geom = "point", shape = 23, size = 3, fill = "white", color = "black") +  # 
  labs(
    title = "Number of Representative Gestures During Pretend Play",
    x = "Group",
    y = "Representative Gestures"
  ) +
  scale_fill_grey(start = 0.4, end = 0.8) +
  theme_minimal(base_size = 12) +
  theme(
    legend.position = "none",
    text = element_text(family = "Times New Roman"),
    plot.title = element_text(hjust = 0.5, face = "bold")
  )
```



Bar Graph of Means for Poportion of Rep Gest

```r
# Create clean bar graph of proportion of representative gestures
pp_gesture_data_ALLyears %>%
  ggplot(aes(x = groupstatus, y = prop_rep, fill = groupstatus)) +
  stat_summary(fun = mean, geom = "bar", width = 0.6, color = "black", alpha = 0.7) +
  stat_summary(fun.data = mean_se, geom = "errorbar", width = 0.2) +
  labs(
    title = NULL,
    x = "Group",
    y = "Proportion of Representative Gestures"
  ) +
  scale_fill_grey(start = 0.4, end = 0.8) +
  scale_y_continuous(labels = scales::percent_format(accuracy = 1)) +  # Display y-axis as percentages
  theme_minimal(base_size = 12) +
  theme(
    legend.position = "none",
    text = element_text(family = "Times New Roman"),
    axis.title.y = element_text(margin = margin(r = 10)),
    plot.title = element_text(hjust = 0.5, face = "bold")
  )
```

```
## Warning: Removed 14 rows containing non-finite outside the scale range
## (`stat_summary()`).
## Removed 14 rows containing non-finite outside the scale range
## (`stat_summary()`).
```



```
# Figure X
# Mean proportion of representative gestures during pretend play by group. Error bars represent ±1 stan
```

OLD CODE

table opton 1

```
# Load required libraries
library(dplyr)
library(gt)

# Create a summary table manually (based on your results so far)
mann_whitney_summary <- tibble::tibble(
  Outcome = c(
    "Total Gestures",
    "Number of Representational Gestures",
    "Proportion of Representational Gestures"
  ),
  W = c(802, 867, 615.5),
  p = c(0.235, 0.036, 0.011),
  r = c(0.13, 0.23, 0.29)
)

# Clean up for APA formatting
mann_whitney_summary <- mann_whitney_summary %>%
  mutate(
    p = ifelse(p < .001, "< .001", sprintf("%.3f", p)),
    r = sprintf("%.2f", r)
  )

# Create the APA-style table
mann_whitney_summary %>%
  gt() %>%
  tab_header(
    title = md("**Table X**"),
    subtitle = md("**Mann-Whitney U Test Results for Gesture Outcomes**")
  ) %>%
  cols_label(
    Outcome = "Outcome Variable",
    W = "W",
    p = "p",
    r = "Effect Size (r)"
  ) %>%
  tab_options(
    table.font.size = "small",
    table.align = "center",
    data_row.padding = px(2)
  )
# Note. W = Wilcoxon rank-sum statistic (equivalent to Mann-Whitney U); r = effect size
```

table opton 2

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

# Step 1: Create summary table
mann_whitney_summary <- tibble::tibble(
  Outcome = c(
    "Total Gestures",
    "Number of Representational Gestures",
    "Proportion of Representational Gestures"
  ),
  W = c(802, 867, 615.5),
  p_numeric = c(0.235, 0.036, 0.011),
  r = c(0.13, 0.23, 0.29)
)

# Step 2: Format nicely for APA
mann_whitney_summary <- mann_whitney_summary %>%
  mutate(
    p = ifelse(p_numeric < .001, "< .001", sprintf("%.3f", p_numeric)),
    r = sprintf("%.2f", r)
  )

# Step 3: Build APA table with bolded significant rows
mann_whitney_summary %>%
  select(Outcome, W, p, r, p_numeric) %>%  # Keep p_numeric for bolding condition
  gt() %>%
```

```r
  tab_header(
    title = md("**Table X**"),
    subtitle = md("**Mann-Whitney U Test Results for Gesture Outcomes**")
  ) %>%
  cols_label(
    Outcome = "Outcome Variable",
    W = "W",
    p = "p",
    r = "Effect Size (r)"
  ) %>%
  tab_style(
    style = cell_text(weight = "bold"),
    locations = cells_body(
      rows = p_numeric < 0.05
    )
  ) %>%
  tab_options(
    table.font.size = "small",
    table.align = "center",
    data_row.padding = px(2)
  ) %>%
  cols_hide(columns = p_numeric)  # Hide helper column p_numeric
```

old Mann-Whtiney for rep gest

```r
# Mann-Whitney U Test (nonparametric equivalent to independent-samples t-test)
wilcox.test(rep_gestures ~ groupstatus, data = pp_gesture_data_ALLyears)

#report medians (which is preferred for nonparametric tests)
pp_gesture_data_ALLyears %>%
  group_by(groupstatus) %>%
  summarise(
    Median_RepGestures = median(rep_gestures, na.rm = TRUE),
    n = n()
  )

# Step 2a: Get N
n1 <- pp_gesture_data_ALLyears %>%
  filter(groupstatus == "TD") %>%
  nrow()

n2 <- pp_gesture_data_ALLyears %>%
  filter(groupstatus == "PL") %>%
  nrow()

N <- n1 + n2

# Step 2b: Approximate Z from p-value
# Two-tailed p-value, so divide by 2
z_approx <- qnorm(mw_result$p.value / 2, lower.tail = FALSE)

# Step 2c: Calculate r
r_effect_size <- z_approx / sqrt(N)
```

```r
# Print it
r_effect_size
```

boxplot of rep gest

```r
library(ggplot2)

ggplot(pp_gesture_data_ALLyears, aes(x = groupstatus, y = rep_gestures)) +

  # Boxplot
  geom_boxplot(fill = "gray80", color = "black", width = 0.6) +

  # Add individual data points
  geom_jitter(width = 0.1, height = 0, alpha = 0.5, size = 2) +

  # Labels
  labs(
    title = "Representational Gestures by Group Status",
    x = "Group Status",
    y = "Number of Representational Gestures"
  ) +

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

split bar graph

```r
# Create binary variable: 0 = No rep gesture, 1 = At least one rep gesture
pp_gesture_data_ALLyears <- pp_gesture_data_ALLyears %>%
  mutate(rep_gesture_present = if_else(rep_gestures > 0, 1, 0))

# Bar plot
ggplot(pp_gesture_data_ALLyears, aes(x = groupstatus, fill = factor(rep_gesture_present))) +
  geom_bar(position = "fill", color = "black") +
  scale_fill_manual(
    values = c("gray80", "black"),
    labels = c("Absent", "Present"),
    name = "Representational Gesture"
  ) +
  labs(
    title = "Proportion of Children Producing Representational Gestures",
    x = "Group Status",
    y = "Proportion"
  ) +
  theme_minimal(base_size = 14, base_family = "Times New Roman") +
  theme(
    panel.border = element_rect(color = "black", fill = NA),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    plot.title = element_text(hjust = 0.5, face = "bold"),
    axis.title = element_text(face = "bold"),
    legend.position = "bottom"
  )
#Figure 1. Proportion of children producing representational gestures during pretend play, by group sta
```

if want figure with individual measures

```r
ggplot(pp_gesture_data_ALLyears, aes(x = groupstatus, y = rep_gestures)) +
  geom_jitter(width = 0.2, height = 0, alpha = 0.7, size = 2, color = "black") +
  labs(
    title = "Representational Gesture Counts by Group",
    x = "Group Status",
    y = "Number of Representational Gestures"
  ) +
  theme_minimal(base_size = 14, base_family = "Times New Roman") +
  theme(
    panel.border = element_rect(color = "black", fill = NA),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    plot.title = element_text(face = "bold", hjust = 0.5),
    axis.title = element_text(face = "bold")
  )
#Figure 2. Individual participant counts of representational gestures during pretend play, grouped by g
```

treat rep gestures as binary

```r
# Make sure rep_gesture_present is coded 0/1 (binary)
pp_gesture_data_ALLyears <- pp_gesture_data_ALLyears %>%
  mutate(rep_gesture_present = if_else(rep_gestures > 0, 1, 0))

# Create a table
table_group_gesture <- table(pp_gesture_data_ALLyears$groupstatus, pp_gesture_data_ALLyears$rep_gesture_

# Run chi-square test
chisq.test(table_group_gesture)
```

split bargraph

```r
library(ggplot2)
library(dplyr)

# Make sure rep_gesture_present is binary (0 = Absent, 1 = Present)
pp_gesture_data_ALLyears <- pp_gesture_data_ALLyears %>%
  mutate(rep_gesture_present = if_else(rep_gestures > 0, 1, 0))
```

```r
# Create the bar plot
ggplot(pp_gesture_data_ALLyears, aes(x = groupstatus, fill = factor(rep_gesture_present))) +

  geom_bar(position = "fill", color = "black") +

  scale_fill_manual(
    values = c("gray80", "black"),  # APA-style colors
    labels = c("Absent", "Present"),  # Legend labels
    name = "Representational Gesture Presence"
  ) +

  labs(
    title = "Proportion of Children Producing Representational Gestures",
    x = "Group Status",
    y = "Proportion"
  ) +

  theme_minimal(base_size = 14, base_family = "Times New Roman") +

  theme(
    panel.border = element_rect(color = "black", fill = NA),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    plot.title = element_text(face = "bold", hjust = 0.5),
    axis.title = element_text(face = "bold"),
    legend.position = "bottom",
    axis.text = element_text(color = "black")
  )
```

1. Conduct ANOVAs for gestures & summarize results

```r
# For total gestures
total_anova <- aov(total_gestures ~ groupstatus, pp_gesture_data_ALLyears)
cat("\nTotal gestures ANOVA:\n")
print(summary(total_anova))

# For representational gestures
rep_anova <- aov(rep_gestures ~ groupstatus, data = pp_gesture_data_ALLyears)
cat("\nRepresentational gestures ANOVA:\n")
print(summary(rep_anova))

# For proportion of representational gestures to total gestures
prop_rep_anova <- aov(prop_rep ~ groupstatus, data = pp_gesture_data_ALLyears)
cat("\nProportion of representational gestures ANOVA:\n")
print(summary(prop_rep_anova))
```

2. Calculate effect sizes

```r
if(require(effectsize)) {
  cat("\nEffect sizes:\n")
  cat("Total gestures: ")
  print(eta_squared(total_anova))
  cat("Proportion representational: ")
  print(eta_squared(prop_rep_anova))
} else {
  # Manual calculation if package not available
  cat("\nEffect sizes calculated manually:\n")
  # Formula for eta-squared: SS_between / SS_total
  summary_total <- summary(total_anova)
  eta_sq_total <- summary_total[[1]]["GroupStatus", "Sum Sq"] /
                 sum(summary_total[[1]][, "Sum Sq"])
  cat("Total gestures eta-squared: ", eta_sq_total, "\n")
  # Repeat for other ANOVAs
}
```

3. Descriptive statistics by group

```
group_stats <- pp_gesture_data_ALLyears %>%
  group_by(groupstatus) %>%
  summarize(
    n = n(),
    total_mean = mean(total_gestures, na.rm = TRUE),
    total_sd = sd(total_gestures, na.rm = TRUE),
    rep_mean = mean(rep_gestures, na.rm = TRUE),
    rep_sd = sd(rep_gestures, na.rm = TRUE),
    prop_rep_mean = mean(prop_rep, na.rm = TRUE),
    prop_rep_sd = sd(prop_rep, na.rm = TRUE)
  )

print("Descriptive statistics by group:")
print(group_stats)
```

4. Visualization of results

```
if(require(tidyverse)) {
  # Raw counts visualization
  counts_long <- pp_gesture_data_ALLyears %>%
    select(groupstatus, total_gestures, rep_gestures) %>%
    pivot_longer(cols = c(total_gestures, rep_gestures),
                names_to = "gesture_type",
                values_to = "count")

  p1 <- ggplot(counts_long, aes(x = gesture_type, y = count, fill = groupstatus)) +
    stat_summary(fun = mean, geom = "bar", position = position_dodge(0.9)) +
    stat_summary(fun.data = function(x) {
      data.frame(y = mean(x, na.rm = TRUE),
                ymin = mean(x, na.rm = TRUE) - sd(x, na.rm = TRUE)/sqrt(sum(!is.na(x))),
                ymax = mean(x, na.rm = TRUE) + sd(x, na.rm = TRUE)/sqrt(sum(!is.na(x))))
    }, geom = "errorbar", width = 0.2, position = position_dodge(0.9)) +
    labs(title = "Mean Gesture Counts During Pretend Play by Group",
         x = "Gesture Type",
         y = "Mean Count",
         fill = "Group Status") +
    theme_minimal() +
    scale_x_discrete(labels = c("total_gestures" = "Total",
                               "rep_gestures" = "Representational"))
  print(p1)

  # Proportions visualization
  props_long <- pp_gesture_data_ALLyears %>%
    select(groupstatus, prop_rep) %>%
    pivot_longer(cols = c(prop_rep),
                names_to = "proportion_type",
                values_to = "proportion")

  p2 <- ggplot(props_long, aes(x = proportion_type, y = proportion, fill = groupstatus)) +
    stat_summary(fun = mean, geom = "bar", position = position_dodge(0.9)) +
    stat_summary(fun.data = function(x) {
      data.frame(y = mean(x, na.rm = TRUE),
                ymin = mean(x, na.rm = TRUE) - sd(x, na.rm = TRUE)/sqrt(sum(!is.na(x))),
                ymax = mean(x, na.rm = TRUE) + sd(x, na.rm = TRUE)/sqrt(sum(!is.na(x))))
    }, geom = "errorbar", width = 0.2, position = position_dodge(0.9)) +
    labs(title = "Proportion of Gesture Types During Pretend Play by Group",
         x = "Proportion Type",
         y = "Mean Proportion",
         fill = "Group Status") +
    theme_minimal() +
    scale_x_discrete(labels = c("prop_rep" = "Representational/Total")) +
    scale_y_continuous(labels = scales::percent)
  print(p2)
}
```

5. Statistical tests to directly compare TD and BI groups

```
# t-tests for each variable (alternative to ANOVA with only two groups)
t_total <- t.test(total_gestures ~ groupstatus, data = pp_gesture_data_ALLyears)
t_rep <- t.test(rep_gestures ~ groupstatus, data = pp_gesture_data_ALLyears)
t_prop_rep <- t.test(prop_rep ~ groupstatus, data = pp_gesture_data_ALLyears)

cat("\nt-test results (direct comparison between groups):\n")
cat("\nTotal gestures:\n")
print(t_total)
cat("\nRepresentational gestures:\n")
print(t_rep)
cat("\nProportion representational:\n")
print(t_prop_rep)
```

6. Calculate Cohen's d effect sizes for t-tests with error handling

```
cat("\nCohen's d effect sizes:\n")

# Function to safely calculate Cohen's d
safe_cohens_d <- function(formula, data) {
  tryCatch({
    res <- cohens_d(formula, data = data)
    return(res)
```

```r
  }, error = function(e) {
    # Extract variable name from formula
    var_name <- as.character(formula)[2]
    # Get means by group
    means <- aggregate(formula, data = data, FUN = mean, na.rm = TRUE)
    sds <- aggregate(formula, data = data, FUN = sd, na.rm = TRUE)

    cat("Error calculating Cohen's d for", var_name, ":\n")
    cat("Group means:", toString(means), "\n")
    cat("Group SDs:", toString(sds), "\n")
    cat("Error message:", e$message, "\n")
    return(NULL)
  })
}

# Apply the safe function to each variable
cat("\nTotal gestures:\n")
print(safe_cohens_d(total_gestures ~ groupstatus, data = pp_gesture_data_ALLyears))

cat("\nRepresentational gestures:\n")
print(safe_cohens_d(rep_gestures ~ groupstatus, data = pp_gesture_data_ALLyears))

cat("\nProportion representational:\n")
print(safe_cohens_d(prop_rep ~ groupstatus, data = pp_gesture_data_ALLyears))
```

Make APA-formatted data visualizations

```r
if(require(tidyverse)) {
  # Set APA theme
  apa_theme <- theme_bw() +
    theme(
      panel.grid.major = element_blank(),
      panel.grid.minor = element_blank(),
      panel.border = element_rect(colour = "black", fill = NA),
      axis.text = element_text(colour = "black", size = 12),
      axis.title = element_text(size = 12, face = "bold"),
      legend.title = element_text(size = 12, face = "bold"),
      legend.text = element_text(size = 12),
      legend.position = "bottom",
      legend.background = element_rect(fill = "white", colour = "black"),
      plot.title = element_text(size = 14, hjust = 0.5, face = "bold"),
      text = element_text(family = "Times New Roman")
    )

  # Raw counts visualization
  counts_long <- pp_gesture_data_ALLyears %>%
    select(groupstatus, total_gestures, rep_gestures) %>%
    pivot_longer(cols = c(total_gestures, rep_gestures),
                names_to = "gesture_type",
                values_to = "count")

  # Renaming GroupStatus labels to be more readable
  counts_long$groupstatus <- factor(counts_long$groupstatus,
                                  levels = c("PL", "TD"),
                                  labels = c("Perinatal Lesions", "Typically Developing"))

  p1 <- ggplot(counts_long, aes(x = gesture_type, y = count, fill = groupstatus)) +
    stat_summary(fun = mean, geom = "bar", position = position_dodge(0.8),
                alpha = 0.8, color = "black", size = 0.2) +
    stat_summary(fun.data = function(x) {
      data.frame(y = mean(x, na.rm = TRUE),
                ymin = mean(x, na.rm = TRUE) - sd(x, na.rm = TRUE)/sqrt(sum(!is.na(x))),
                ymax = mean(x, na.rm = TRUE) + sd(x, na.rm = TRUE)/sqrt(sum(!is.na(x))))
    }, geom = "errorbar", width = 0.2, position = position_dodge(0.8)) +
    labs(title = "Figure 1",
         subtitle = "Mean Gesture Counts During Pretend Play by Group",
         x = "Gesture Type",
         y = "Mean Count",
         fill = "Group") +
    scale_x_discrete(labels = c("total_gestures" = "Total",
                               "rep_gestures" = "Representational")) +
    scale_fill_grey(start = 0.4, end = 0.8) +
    apa_theme +
    theme(plot.subtitle = element_text(size = 12, hjust = 0.5))

  print(p1)

  # Save high-resolution figure
  ggsave("Figure_1_Gesture_Counts.tiff", p1, width = 7, height = 5, dpi = 300)

  # Proportions visualization
  props_long <- pp_gesture_data_ALLyears %>%
    select(groupstatus, prop_rep) %>%
    pivot_longer(cols = c(prop_rep),
                names_to = "proportion_type",
                values_to = "proportion")

  # Renaming GroupStatus labels
  props_long$groupstatus <- factor(props_long$groupstatus,
                                  levels = c("PL", "TD"),
                                  labels = c("Perinatal Lesions", "Typically Developing"))

  p2 <- ggplot(props_long, aes(x = proportion_type, y = proportion, fill = groupstatus)) +
    stat_summary(fun = mean, geom = "bar", position = position_dodge(0.8),
                alpha = 0.8, color = "black", size = 0.2) +
    stat_summary(fun.data = function(x) {
      data.frame(y = mean(x, na.rm = TRUE),
                ymin = mean(x, na.rm = TRUE) - sd(x, na.rm = TRUE)/sqrt(sum(!is.na(x))),
                ymax = mean(x, na.rm = TRUE) + sd(x, na.rm = TRUE)/sqrt(sum(!is.na(x))))
    }, geom = "errorbar", width = 0.2, position = position_dodge(0.8)) +
    labs(title = "Figure 2",
         subtitle = "Proportion of Gesture Types During Pretend Play by Group",
         x = "Proportion Type",
         y = "Mean Proportion",
         fill = "Group") +
    scale_x_discrete(labels = c("prop_rep" = "Representational/Total")) +
    scale_y_continuous(labels = scales::percent,
                      breaks = seq(0, 0.25, 0.05)) +
    scale_fill_grey(start = 0.4, end = 0.8) +
    apa_theme +
    theme(plot.subtitle = element_text(size = 12, hjust = 0.5))

  print(p2)

  # Save high-resolution figure
  ggsave("Figure_2_Gesture_Proportions.tiff", p2, width = 7, height = 5, dpi = 300)
}
```

Is rep_gestures positively correlated with total_gestures (overall gestures produced during pretend play)?

```
cor.test(pp_gesture_data_ALLyears$rep_gestures, pp_gesture_data_ALLyears$total_gestures)
```

boxplot of representational gesture presence & narrative score

```
# First: recode rep_gestures as a factor for clean labeling
pp_gesture_data_ALLyears <- pp_gesture_data_ALLyears %>%
  mutate(rep_gesture_factor = factor(rep_gesture_present, levels = c(0,1), labels = c("Absent", "Presen

# Plot
ggplot(pp_gesture_data_ALLyears, aes(x = rep_gesture_factor, y = max_avg)) +
  geom_boxplot(fill = "gray80", color = "black") +
  stat_summary(fun = mean, geom = "point", size = 3, color = "black", position = position_dodge(width=0
  labs(
    x = "Representational Gesture",
    y = "Narrative Structure Score",
    title = "Narrative Structure by Representational Gesture Presence"
  ) +
  theme_minimal(base_size = 14, base_family = "Times New Roman") +
  theme(
    panel.border = element_rect(color = "black", fill = NA),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    plot.title = element_text(hjust = 0.5, face = "bold"),
    axis.title = element_text(face = "bold")
  )
```

```
library(ggplot2)
library(dplyr)

# Recode rep_gestures as a factor for clean group labels
pp_gesture_data_ALLyears <- pp_gesture_data_ALLyears %>%
  mutate(rep_gesture_factor = factor(rep_gesture_present, levels = c(0, 1),
                                     labels = c("Absent", "Present")))

# Bar graph with error bars (mean ± SE)
ggplot(pp_gesture_data_ALLyears, aes(x = rep_gesture_factor, y = max_avg)) +

  # Bar for mean narrative scores
  stat_summary(fun = mean, geom = "bar", fill = "gray80", color = "black", width = 0.6) +

  # Error bars (± standard error)
  stat_summary(fun.data = mean_se, geom = "errorbar", width = 0.2, color = "black", size = 0.8) +

  # Points for individual data points (optional, adds transparency)
  geom_jitter(width = 0.1, height = 0, size = 2, alpha = 0.5) +

  # Labels
  labs(
    title = "Narrative Structure by Representational Gesture Presence",
    x = "Representational Gesture Presence",
    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"
  )

#Figure 1
#Mean narrative structure scores for children who did versus did not produce representational gestures
```

```
pp_gesture_data_ALLyears %>%
  group_by(rep_gesture_factor) %>%
  summarise(
    Mean_Narrative = mean(max_avg, na.rm = TRUE),
    SD_Narrative = sd(max_avg, na.rm = TRUE),
    n = n()
  )
```

```
# Identify participants with missing rep_gestures
pp_gesture_data_ALLyears %>%
  filter(is.na(rep_gestures)) %>%
  select(participant_id, total_gestures, rep_gestures, everything())

# List all unique values
unique(pp_gesture_data_ALLyears$rep_gestures)

# Find participants with rep_gestures not equal to 0 or 1
pp_gesture_data_ALLyears %>%
  filter(!(rep_gestures %in% c(0, 1))) %>%
  select(participant_id, rep_gestures)
```