PretendPlay_Gesture_Analysis

Kristen Johnson

2025-04-02

```
##
## The downloaded binary packages are in
##  /var/folders/w3/z47w_pmn3h190rzxwvk8l5w40000gn/T//Rtmpm2f6HP/downloaded_packages

##
## The downloaded binary packages are in
##  /var/folders/w3/z47w_pmn3h190rzxwvk8l5w40000gn/T//Rtmpm2f6HP/downloaded_packages

##
## The downloaded binary packages are in
##  /var/folders/w3/z47w_pmn3h190rzxwvk8l5w40000gn/T//Rtmpm2f6HP/downloaded_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 error
## 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
```

My data is in a dataframe called 'merged_[timepoint]' with columns:

- GroupStatus: Factor with levels "TD" and "PL"

- gesture_all: Count of all gestures

- gesture_rep: Count of representational gestures

- gesture_icon: Count of iconic gestures

- C-wpu: Language ability measure

- c_pret: Frequency of pretend play instances

1. Conduct ANOVAs for each gesture type & summarize results

```
# read in data frame that is just gesture during instance of pretend play
pretend_data_H8 <- read.csv("~/KristenWorkingDirectory/Play_Narrative/PN_Datasets/Pretend_Play/CSVs_of_

# Aggregate by child
child_summary <- pretend_data_H8 %>%
  group_by(participant_id, GroupStatus) %>% # Assuming you have a child_id variable
  summarize(
    total_gestures = sum(gesture_all),
    rep_gestures = sum(gesture_rep),
    icon_gestures = sum(gesture_icon),
    prop_rep = sum(gesture_rep) / sum(gesture_all),
    prop_icon = sum(gesture_icon) / sum(gesture_all)
  )
```

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

```
# For total gestures
total_anova <- aov(total_gestures ~ GroupStatus, data = child_summary)
cat("\nTotal gestures ANOVA:\n")
```

```
##
## Total gestures ANOVA:
```

```
print(summary(total_anova))
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## GroupStatus  1   38.5   38.51     1.8  0.194
## Residuals   21  449.2   21.39
```

```
# For representational gestures
rep_anova <- aov(rep_gestures ~ GroupStatus, data = child_summary)
cat("\nRepresentational gestures ANOVA:\n")
```

```
##
## Representational gestures ANOVA:
```

```r
print(summary(rep_anova))
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## GroupStatus  1  0.419  0.4188   0.665  0.424
## Residuals   21 13.233  0.6302
```

```r
# For iconic gestures
icon_anova <- aov(icon_gestures ~ GroupStatus, data = child_summary)
cat("\nIconic gestures ANOVA:\n")
```

```
##
## Iconic gestures ANOVA:
```

```r
print(summary(icon_anova))
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## GroupStatus  1  0.117  0.1174   0.188  0.669
## Residuals   21 13.100  0.6238
```

```r
# For proportion of representational gestures to total gestures
prop_rep_anova <- aov(prop_rep ~ GroupStatus, data = child_summary)
cat("\nProportion of representational gestures ANOVA:\n")
```

```
##
## Proportion of representational gestures ANOVA:
```

```r
print(summary(prop_rep_anova))
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## GroupStatus  1 0.0676 0.06765   1.585  0.227
## Residuals   15 0.6403 0.04269
## 6 observations deleted due to missingness
```

```r
#for proportion of iconic gestures to total gestures
prop_icon_anova <- aov(prop_icon ~ GroupStatus, data = child_summary)
cat("\nProportion of iconic gestures ANOVA:\n")
```

```
##
## Proportion of iconic gestures ANOVA:
```

```r
print(summary(prop_icon_anova))
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## GroupStatus  1 0.0402 0.04015     0.9  0.358
## Residuals   15 0.6690 0.04460
## 6 observations deleted due to missingness
```

2. Calculate effect sizes

```r
if(require(effectsize)) {
  cat("\nEffect sizes:\n")
  cat("Total gestures: ")
  print(eta_squared(total_anova))
  cat("Representational gestures: ")
  print(eta_squared(rep_anova))
  cat("Iconic gestures: ")
  print(eta_squared(icon_anova))
  cat("Proportion representational: ")
  print(eta_squared(prop_rep_anova))
  cat("Proportion iconic: ")
  print(eta_squared(prop_icon_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
}
```

```
##
## Effect sizes:
## Total gestures: GroupStatus
##  0.07894752
## Representational gestures: GroupStatus
##  0.03067941
## Iconic gestures: GroupStatus
## 0.008881579
## Proportion representational: GroupStatus
##  0.09555083
## Proportion iconic: GroupStatus
##  0.05662302
```

3. Descriptive statistics by group

```
group_stats <- child_summary %>%
  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),
    icon_mean = mean(icon_gestures, na.rm = TRUE),
    icon_sd = sd(icon_gestures, na.rm = TRUE),
    prop_rep_mean = mean(prop_rep, na.rm = TRUE),
    prop_rep_sd = sd(prop_rep, na.rm = TRUE),
    prop_icon_mean = mean(prop_icon, na.rm = TRUE),
    prop_icon_sd = sd(prop_icon, na.rm = TRUE)
  )

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

```
## [1] "Descriptive statistics by group:"
```

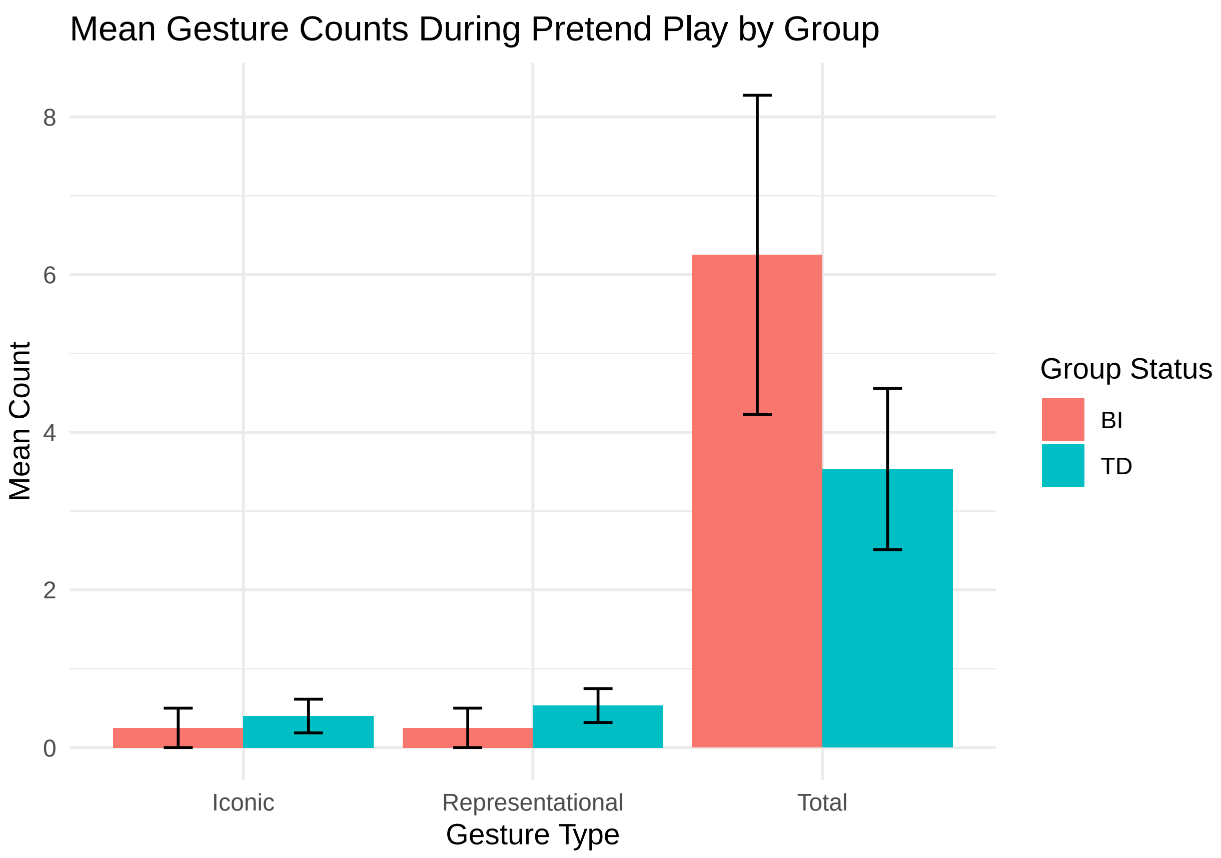
```r
print(group_stats)
```

```
## # A tibble: 2 x 12
##   GroupStatus     n total_mean total_sd rep_mean rep_sd icon_mean icon_sd
##   <chr>       <int>      <dbl>    <dbl>    <dbl>  <dbl>     <dbl>   <dbl>
## 1 BI              8       6.25     5.73    0.25   0.707      0.25   0.707
## 2 TD             15       3.53     3.96    0.533  0.834      0.4    0.828
## # i 4 more variables: prop_rep_mean <dbl>, prop_rep_sd <dbl>,
## #   prop_icon_mean <dbl>, prop_icon_sd <dbl>
```

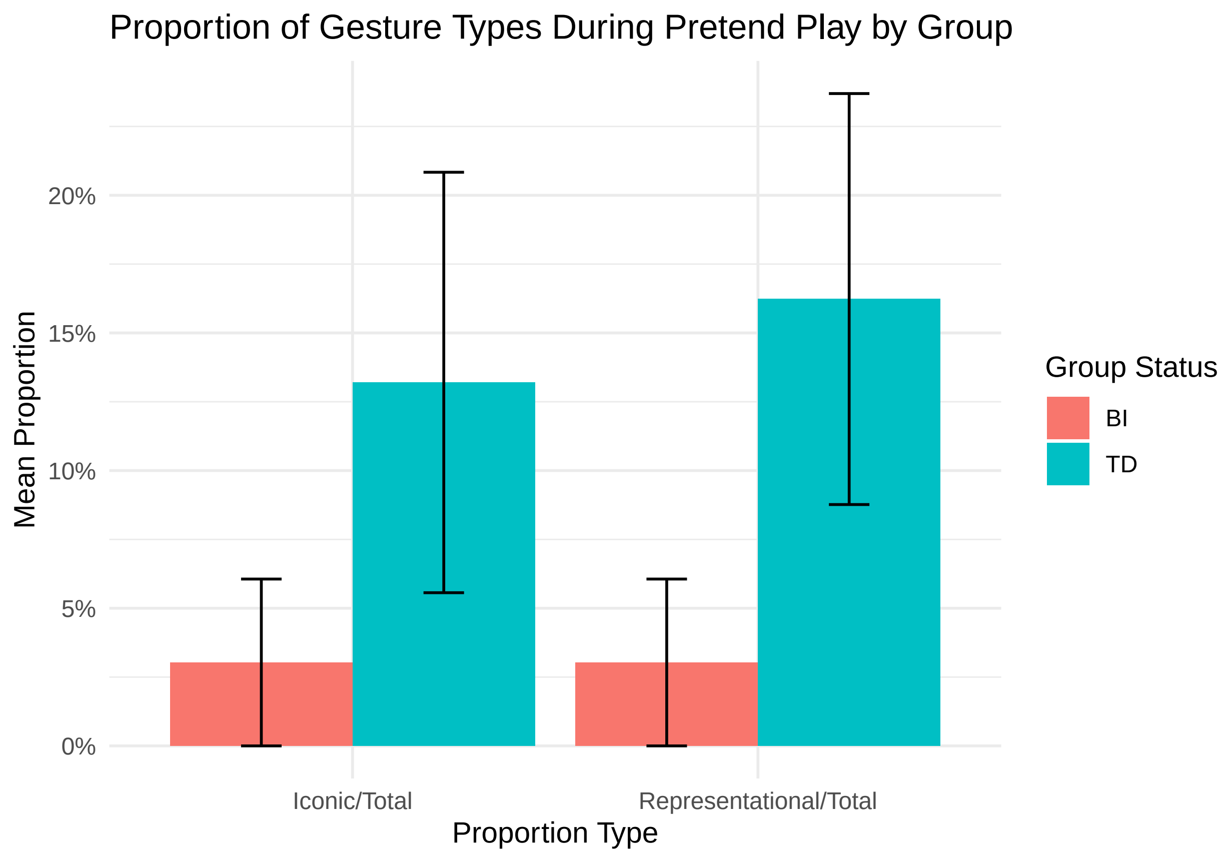
4. Visualization of results

```r
if(require(tidyverse)) {
  # Raw counts visualization
  counts_long <- child_summary %>%
    select(GroupStatus, total_gestures, rep_gestures, icon_gestures) %>%
    pivot_longer(cols = c(total_gestures, rep_gestures, icon_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",
                                "icon_gestures" = "Iconic"))
  print(p1)

  # Proportions visualization
  props_long <- child_summary %>%
    select(GroupStatus, prop_rep, prop_icon) %>%
    pivot_longer(cols = c(prop_rep, prop_icon),
                 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",
                                "prop_icon" = "Iconic/Total")) +
    scale_y_continuous(labels = scales::percent)
  print(p2)
}
```

```
## Adding missing grouping variables: `participant_id`
```



```
## Adding missing grouping variables: `participant_id`

## Warning: Removed 12 rows containing non-finite outside the scale range
## (`stat_summary()`).
## Removed 12 rows containing non-finite outside the scale range
## (`stat_summary()`).
```



5. Statistical tests to directly compare TD and BI groups

```r
# t-tests for each variable (alternative to ANOVA with only two groups)
t_total <- t.test(total_gestures ~ GroupStatus, data = child_summary)
t_rep <- t.test(rep_gestures ~ GroupStatus, data = child_summary)
t_icon <- t.test(icon_gestures ~ GroupStatus, data = child_summary)
t_prop_rep <- t.test(prop_rep ~ GroupStatus, data = child_summary)
t_prop_icon <- t.test(prop_icon ~ GroupStatus, data = child_summary)

cat("\nt-test results (direct comparison between groups):\n")
```

```
##
## t-test results (direct comparison between groups):
```

```r
cat("\nTotal gestures:\n")
```

```
##
## Total gestures:
```

```r
print(t_total)
```

```
##
##  Welch Two Sample t-test
##
## data:  total_gestures by GroupStatus
## t = 1.1977, df = 10.683, p-value = 0.2569
## alternative hypothesis: true difference in means between group BI and group TD is not equal to 0
## 95 percent confidence interval:
##  -2.293684  7.727017
## sample estimates:
## mean in group BI mean in group TD
##         6.250000         3.533333
```

```
cat("\nRepresentational gestures:\n")
```

```
##
## Representational gestures:
```

```
print(t_rep)
```

```
##
##  Welch Two Sample t-test
##
## data:  rep_gestures by GroupStatus
## t = -0.85879, df = 16.653, p-value = 0.4027
## alternative hypothesis: true difference in means between group BI and group TD is not equal to 0
## 95 percent confidence interval:
##  -0.9805173  0.4138507
## sample estimates:
## mean in group BI mean in group TD
##        0.2500000        0.5333333
```

```
cat("\nIconic gestures:\n")
```

```
##
## Iconic gestures:
```

```
print(t_icon)
```

```
##
##  Welch Two Sample t-test
##
## data:  icon_gestures by GroupStatus
## t = -0.45598, df = 16.556, p-value = 0.6543
## alternative hypothesis: true difference in means between group BI and group TD is not equal to 0
## 95 percent confidence interval:
##  -0.8454635  0.5454635
## sample estimates:
## mean in group BI mean in group TD
##             0.25             0.40
```

```
cat("\nProportion representational:\n")
```

```
##
## Proportion representational:
```

```
print(t_prop_rep)
```

```
##
##  Welch Two Sample t-test
##
## data:  prop_rep by GroupStatus
## t = -1.6387, df = 12.869, p-value = 0.1255
## alternative hypothesis: true difference in means between group BI and group TD is not equal to 0
## 95 percent confidence interval:
```

```
##  -0.30620933  0.04220566
## sample estimates:
## mean in group BI mean in group TD
##       0.03030303       0.16230487
```

```r
cat("\nProportion iconic:\n")
```

```
##
## Proportion iconic:
```

```r
print(t_prop_icon)
```

```
##
##  Welch Two Sample t-test
##
## data:  prop_icon by GroupStatus
## t = -1.2379, df = 12.764, p-value = 0.238
## alternative hypothesis: true difference in means between group BI and group TD is not equal to 0
## 95 percent confidence interval:
##  -0.27951893  0.07612132
## sample estimates:
## mean in group BI mean in group TD
##       0.03030303       0.13200184
```

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

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

```
##
## Cohen's d effect sizes:
```

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

```
##
```

```
## Total gestures:
```

```
print(safe_cohens_d(total_gestures ~ GroupStatus, data = child_summary))
```

```
## Error calculating Cohen's d for total_gestures :
## Group means: c("BI", "TD"), c(6.25, 3.53333333333333)
## Group SDs: c("BI", "TD"), c(5.72588109252317, 3.9617216074881)
## Error message: In argument: `data = map(.data$data, .f, ...)`.
## NULL
```

```
cat("\nRepresentational gestures:\n")
```

```
##
## Representational gestures:
```

```
print(safe_cohens_d(rep_gestures ~ GroupStatus, data = child_summary))
```

```
## Error calculating Cohen's d for rep_gestures :
## Group means: c("BI", "TD"), c(0.25, 0.533333333333333)
## Group SDs: c("BI", "TD"), c(0.707106781186548, 0.833809387832792)
## Error message: In argument: `data = map(.data$data, .f, ...)`.
## NULL
```

```
cat("\nIconic gestures:\n")
```

```
##
## Iconic gestures:
```

```
print(safe_cohens_d(icon_gestures ~ GroupStatus, data = child_summary))
```

```
## Error calculating Cohen's d for icon_gestures :
## Group means: c("BI", "TD"), c(0.25, 0.4)
## Group SDs: c("BI", "TD"), c(0.707106781186548, 0.828078671210825)
## Error message: In argument: `data = map(.data$data, .f, ...)`.
## NULL
```

```
cat("\nProportion representational:\n")
```

```
##
## Proportion representational:
```

```
print(safe_cohens_d(prop_rep ~ GroupStatus, data = child_summary))
```

```
## Error calculating Cohen's d for prop_rep :
## Group means: c("BI", "TD"), c(0.0303030303030303, 0.162304866850321)
## Group SDs: c("BI", "TD"), c(0.0742269619025206, 0.247544545169228)
## Error message: In argument: `data = map(.data$data, .f, ...)`.
## NULL
```

```
cat("\nProportion iconic:\n")
```

```
##
## Proportion iconic:
```

```
print(safe_cohens_d(prop_icon ~ GroupStatus, data = child_summary))
```

```
## Error calculating Cohen's d for prop_icon :
## Group means: c("BI", "TD"), c(0.0303030303030303, 0.132001836547291)
## Group SDs: c("BI", "TD"), c(0.0742269619025206, 0.253266817497245)
## Error message: In argument: `data = map(.data$data, .f, ...)`.
## NULL
```