

Mixed ANOVA Analysis

Your Name

2025-06-03

Contents

Data Visualization	3
Mixed ANOVA Analysis	6
Post-hoc Tests	7
Hierarchical Follow-up	8
Summary	9
Session Info	9

```
# Set path
path_in <- "/mnt/data_dump/pixelstress/4_results"

# Read CSV file
df <- read.csv(file.path(path_in, "stats_table.csv"))

# Check if df exists and is a data frame
if (!exists("df") || !is.data.frame(df)) {
  stop("Please load your dataframe and name it 'df' before running this script.
       Example: df <- read.csv('your_data.csv')")
}

# Check if required columns exist
required_cols <- c("id", "group", "feedback", "stage", "rt", "accuracy", "cnv_Fz", "cnv_Cz")
missing_cols <- setdiff(required_cols, names(df))
if (length(missing_cols) > 0) {
  stop("Missing required columns: ", paste(missing_cols, collapse = ", "))
}

# Ensure factors are properly coded
df$id <- as.factor(df$id)
df$group <- as.factor(df$group)
df$feedback <- as.factor(df$feedback)
df$stage <- as.factor(df$stage)

# Display data structure
str(df)

## 'data.frame':   390 obs. of  17 variables:
##  $ level_0      : int   6 7 8 9 10 11 12 13 14 15 ...
##  $ index        : int   0 1 2 3 4 5 0 1 2 3 ...
##  $ stage        : Factor w/ 2 levels "end","start": 1 1 1 2 2 2 1 1 1 2 ...
##  $ feedback     : Factor w/ 3 levels "above","below",...: 1 2 3 1 2 3 1 2 3 1 ...
##  $ id           : Factor w/ 65 levels "7","8","9","10",...: 30 30 30 30 30 30 54 ...
```

```
## $ group : Factor w/ 2 levels "ctrl","exp": 1 1 1 1 1 1 2 2 2 2 ...
## $ rt : num 330 359 339 446 335 ...
## $ rt_resint : num 246 272 255 372 255 ...
## $ rt_residuals : num -31.75 -5.81 -23.32 93.81 -22.63 ...
## $ n_trials : int 62 52 113 51 59 94 62 61 113 58 ...
## $ accuracy : num 0.939 0.881 0.863 0.911 0.967 ...
## $ combined : chr "ctrl above end" "ctrl below end" "ctrl close end" "ctrl ...
## $ cnv_Fz : num 2.86 1.9 2.4 3.57 3.23 ...
## $ cnv_Cz : num -2.194 -2.141 -2.197 0.602 -1.053 ...
## $ midfrontal_theta_target_FCz_cross: num 3.53 3.44 3.63 2.92 3.01 ...
## $ midfrontal_theta_target_Fz : num 3.33 3.38 3.79 2.74 3.2 ...
## $ midfrontal_theta_target_Cz : num 3.63 3.41 3.38 2.96 2.68 ...
```

```
head(df)
```

```
## level_0 index stage feedback id group rt rt_resint rt_residuals
## 1 6 0 end above 42 ctrl 329.5645 246.2250 -31.7503921
## 2 7 1 end below 42 ctrl 359.2115 272.1690 -5.8063627
## 3 8 2 end close 42 ctrl 338.6283 254.6580 -23.3173902
## 4 9 3 start above 42 ctrl 445.7059 371.7884 93.8130699
## 5 10 4 start below 42 ctrl 334.8475 255.3422 -22.6331879
## 6 11 5 start close 42 ctrl 361.5532 277.9596 -0.0157878
## n_trials accuracy combined cnv_Fz cnv_Cz
## 1 62 0.9393939 ctrl above end 2.863027 -2.1940602
## 2 52 0.8813559 ctrl below end 1.901565 -2.1410654
## 3 113 0.8625954 ctrl close end 2.400586 -2.1971056
## 4 51 0.9107143 ctrl above start 3.570397 0.6022234
## 5 59 0.9672131 ctrl below start 3.228628 -1.0532334
## 6 94 0.9038462 ctrl close start 3.807541 -2.3640199
## midfrontal_theta_target_FCz_cross midfrontal_theta_target_Fz
## 1 3.525576 3.333626
## 2 3.435211 3.383768
## 3 3.632458 3.787888
## 4 2.921590 2.739293
## 5 3.011373 3.202209
## 6 3.220677 2.981376
## midfrontal_theta_target_Cz
## 1 3.633637
## 2 3.407104
## 3 3.377117
## 4 2.959095
## 5 2.682940
## 6 3.305127
```

```
# Display factor levels
```

```
cat("Factor levels:
")
```

```
## Factor levels:
```

```
cat("Group levels:", levels(df$group), "
")
```

```
## Group levels: ctrl exp
```

```
cat("Feedback levels:", levels(df$feedback), "
")
```

```
## Feedback levels: above below close
cat("Stage levels:", levels(df$stage), "
")

## Stage levels: end start
cat("Number of subjects:", length(unique(df$id)), "
")

## Number of subjects: 65
```

Data Visualization

```
# List of dependent variables
dvs <- c("rt", "accuracy", "cnv_Fz", "cnv_Cz")

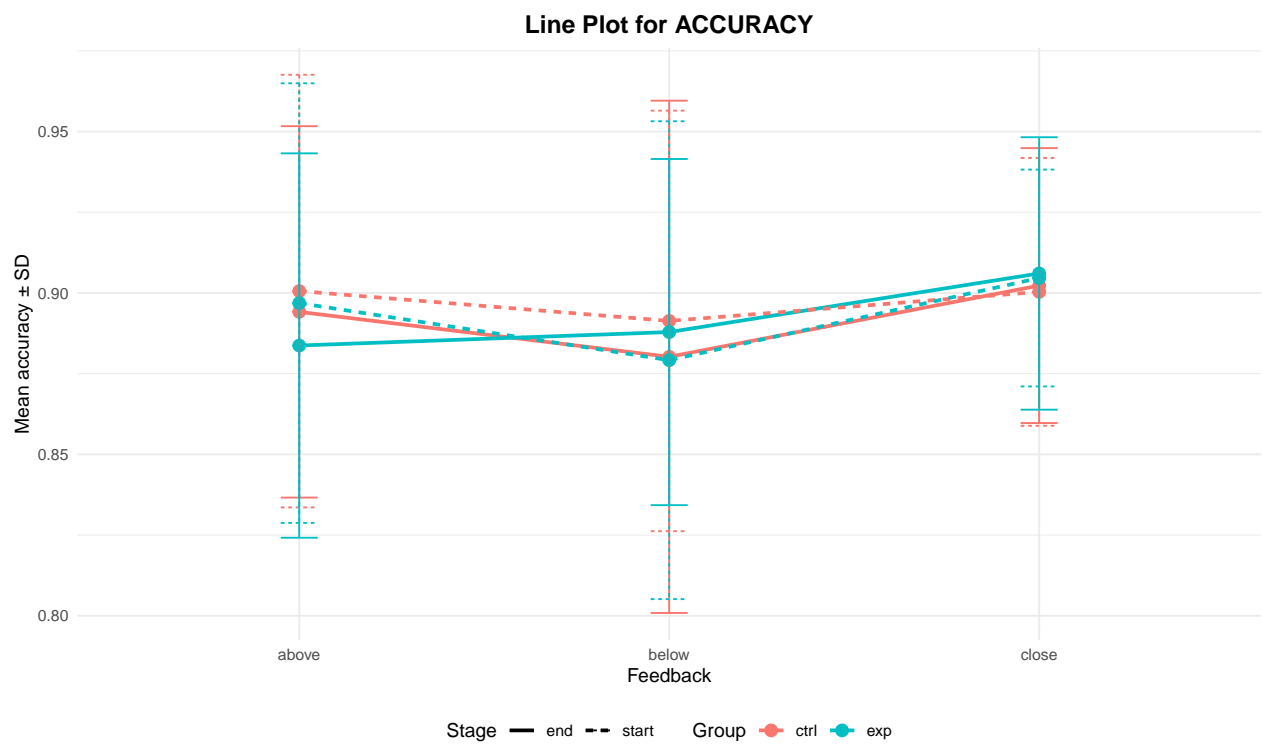
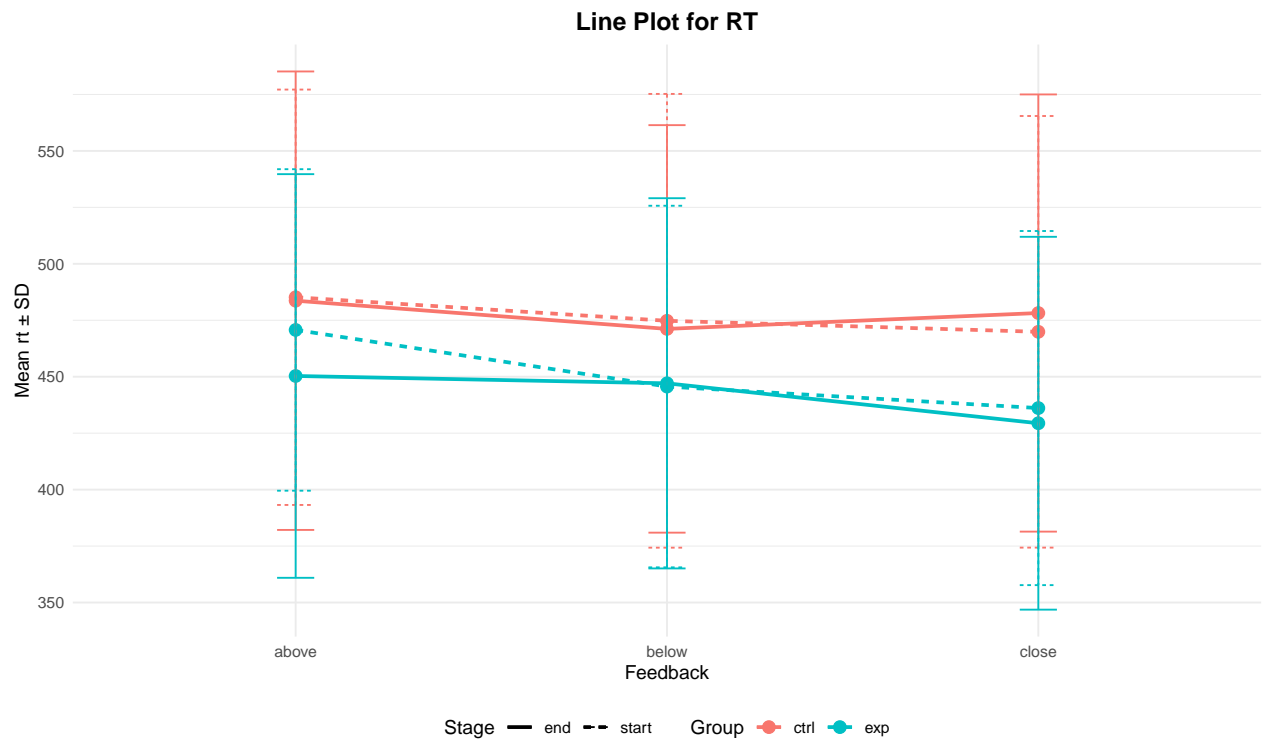
# Function to create line plots
create_lineplot <- function(dv) {
  plot_data <- df %>%
    group_by(group, stage, feedback) %>%
    summarise(
      mean_val = mean(!sym(dv), na.rm = TRUE),
      sd_val = sd(!sym(dv), na.rm = TRUE),
      n = n(),
      se_val = sd_val / sqrt(n),
      .groups = 'drop'
    )

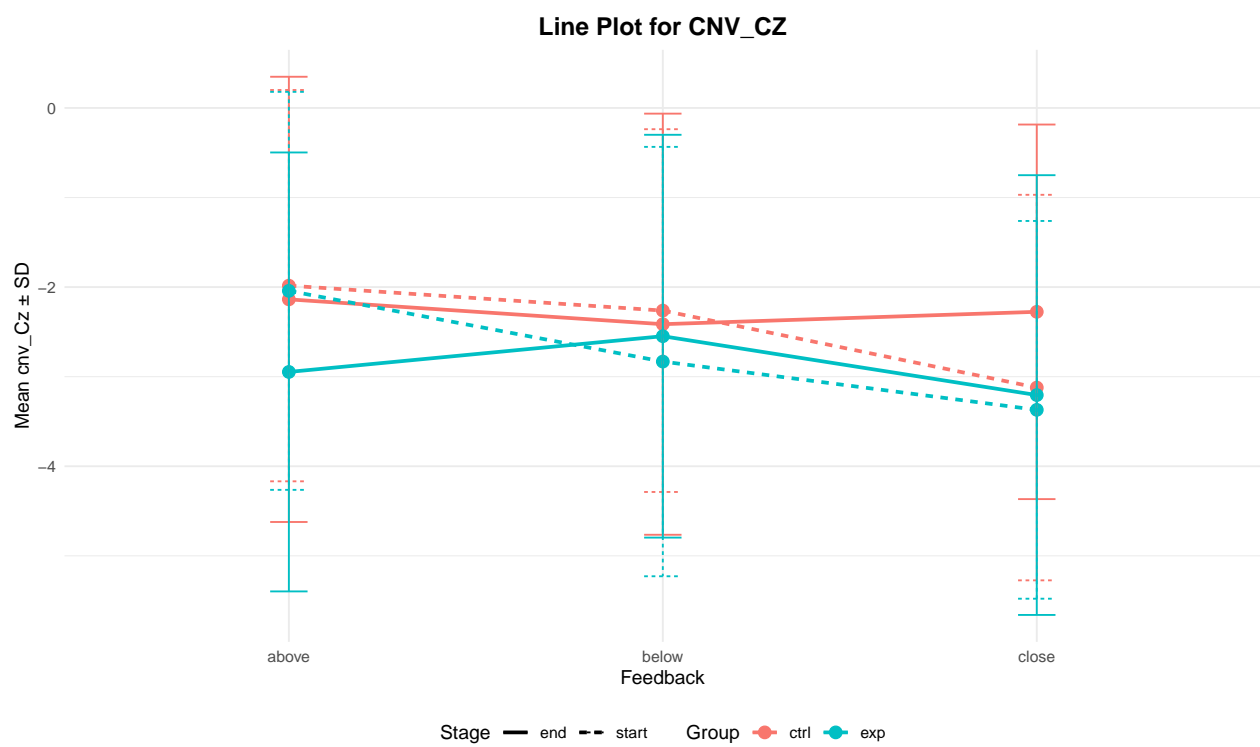
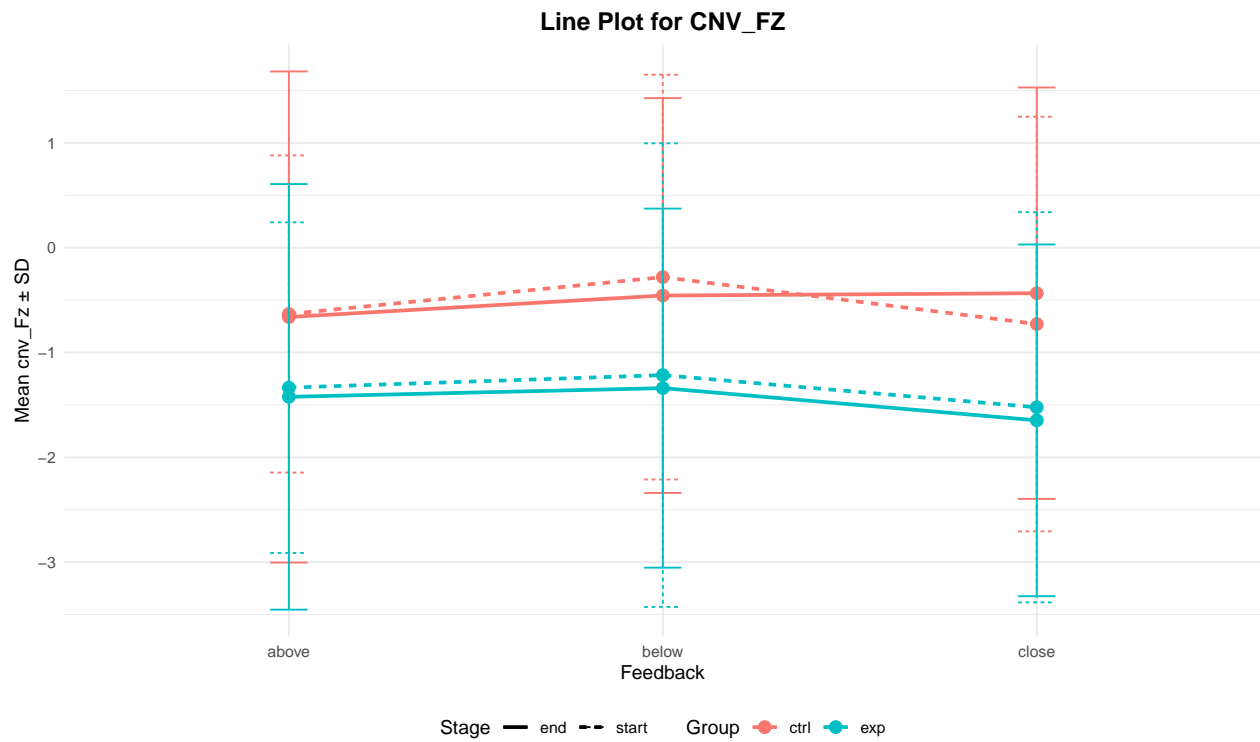
  p <- ggplot(plot_data, aes(x = feedback, y = mean_val, color = group, linetype = stage)) +
    geom_line(aes(group = interaction(group, stage)), size = 1) +
    geom_point(size = 3) +
    geom_errorbar(aes(ymin = mean_val - sd_val, ymax = mean_val + sd_val),
      width = 0.1, size = 0.5) +
    labs(title = paste("Line Plot for", toupper(dv)),
      x = "Feedback",
      y = paste("Mean", dv, "± SD"),
      color = "Group",
      linetype = "Stage") +
    theme_minimal() +
    theme(plot.title = element_text(hjust = 0.5, size = 14, face = "bold"),
      legend.position = "bottom")

  return(p)
}

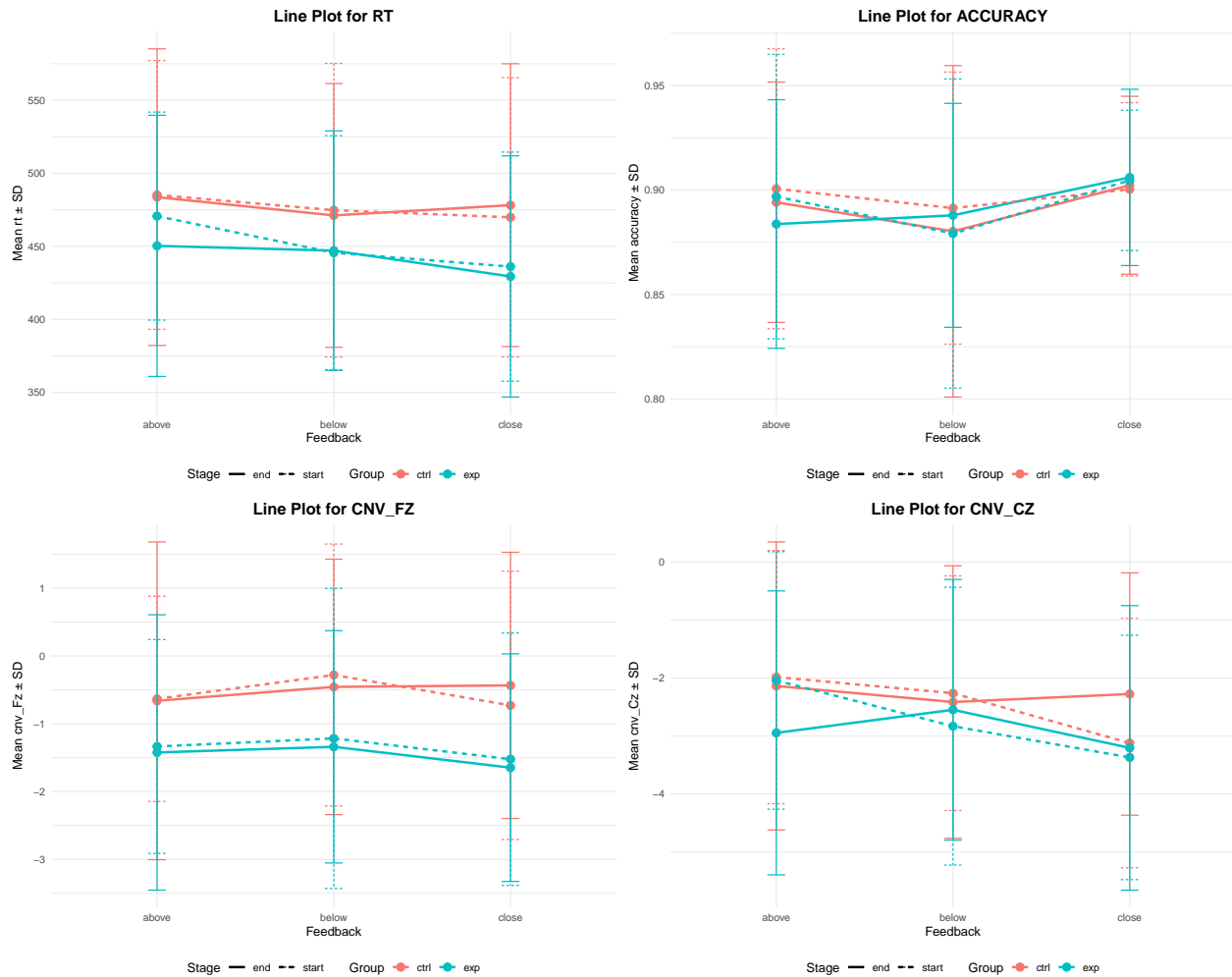
# Create plots
plots <- map(dvs, create_lineplot)
names(plots) <- dvs

# Display plots
walk2(plots, names(plots), ~{
  print(.x)
  cat("\n\n")
})
```





```
combined_plot <- wrap_plots(plots, ncol = 2)
print(combined_plot)
```



Mixed ANOVA Analysis

```
run_mixed_anova <- function(dv) {
  cat(rep("=", 60), "\n")
  cat("MIXED ANOVA RESULTS FOR:", toupper(dv), "\n")
  cat(rep("=", 60), "\n")

  result <- tryCatch({
    anova_result <- ezANOVA(
      data = df,
      dv = dv,
      wid = "id",
      between = "group",
      within = c("stage", "feedback"),
      type = 3,
      detailed = TRUE,
      return_aov = TRUE
    )

    print(anova_result$ANOVA)
  })
}
```

```

    if (!is.null(anova_result$`Mauchly's Test for Sphericity`)) {
      cat("\nMauchly's Test for Sphericity:\n")
      print(anova_result$`Mauchly's Test for Sphericity`)

      if (!is.null(anova_result$`Sphericity Corrections`)) {
        cat("\nSphericity Corrections:\n")
        print(anova_result$`Sphericity Corrections`)
      }
    }

    return(anova_result)
  }, error = function(e) {
    cat("ANOVA failed for", dv, ":", conditionMessage(e), "\n")
    return(NULL)
  })

  return(result)
}

```

```
anova_results <- map(dvs, run_mixed_anova)
```

```

## = = = = =
## MIXED ANOVA RESULTS FOR: RT
## = = = = =
## ANOVA failed for rt : "dv" is not a variable in the data frame provided.
## = = = = =
## MIXED ANOVA RESULTS FOR: ACCURACY
## = = = = =
## ANOVA failed for accuracy : "dv" is not a variable in the data frame provided.
## = = = = =
## MIXED ANOVA RESULTS FOR: CNV_FZ
## = = = = =
## ANOVA failed for cnv_Fz : "dv" is not a variable in the data frame provided.
## = = = = =
## MIXED ANOVA RESULTS FOR: CNV_CZ
## = = = = =
## ANOVA failed for cnv_Cz : "dv" is not a variable in the data frame provided.
names(anova_results) <- dvs

```

Post-hoc Tests

```

run_feedback_posthoc <- function(dv, anova_result) {
  cat("\n" , rep("=", 50), "\n")
  cat("POST-HOC TESTS FOR FEEDBACK -", toupper(dv), "\n")
  cat(rep("=", 50), "\n")

  feedback_p <- anova_result$ANOVA$p[anova_result$ANOVA$Effect == "feedback"]

  if (length(feedback_p) > 0 && feedback_p < 0.05) {
    cat("Feedback main effect is significant (p =", round(feedback_p, 4), ")\n")
    cat("Running pairwise comparisons...\n\n")

    model <- aov(as.formula(paste(dv, "~ group * stage * feedback + Error(id/(stage*feedback))")),

```

```

        data = df)

emm <- emmeans(model, ~ feedback)
pairs <- pairs(emm, adjust = "bonferroni")

print(emm)
cat("\nPairwise comparisons:\n")
print(pairs)

} else {
  cat("Feedback main effect is not significant (p =",
      ifelse(length(feedback_p) > 0, round(feedback_p, 4), "NA"), ")\n")
  cat("Skipping post-hoc tests.\n")
}
}

iwalk(anova_results[!sapply(anova_results, is.null)], run_feedback_posthoc)

```

Hierarchical Follow-up

```

run_hierarchical_followup <- function(dv, anova_result) {
  cat("\n", rep("=", 60), "\n")
  cat("HIERARCHICAL FOLLOW-UP TESTS FOR:", toupper(dv), "\n")
  cat(rep("=", 60), "\n")

  anova_table <- anova_result$ANOVA

  three_way_p <- anova_table$p[anova_table$Effect == "group:stage:feedback"]

  if (length(three_way_p) > 0 && three_way_p < 0.05) {
    cat("3-way interaction is significant (p =", round(three_way_p, 4), ")\n")
    cat("Running follow-up ANOVAs for each GROUP level...\n\n")

    for (grp in levels(df$group)) {
      cat("---- GROUP:", grp, "----\n")
      subset_data <- df[df$group == grp, ]

      follow_up <- ezANOVA(
        data = subset_data,
        dv = dv,
        wid = "id",
        within = c("stage", "feedback"),
        type = 3,
        detailed = TRUE
      )

      print(follow_up$ANOVA)

      stage_feedback_p <- follow_up$ANOVA$p[follow_up$ANOVA$Effect == "stage:feedback"]

      if (length(stage_feedback_p) > 0 && stage_feedback_p < 0.05) {
        for (stg in levels(df$stage)) {
          cat("---- GROUP:", grp, ", STAGE:", stg, "----\n")

```



```

subset_data2 <- subset_data[subset_data$stage == stg, ]

follow_up2 <- ezANOVA(
  data = subset_data2,
  dv = dv,
  wid = "id",
  within = "feedback",
  type = 3,
  detailed = TRUE
)

print(follow_up2$ANOVA)
}
}
}
}
}

iwalk(anova_results[!sapply(anova_results, is.null)], run_hierarchical_followup)

```

Summary

```

cat("Analysis completed for all dependent variables:\n")

## Analysis completed for all dependent variables:
cat("- Line plots created showing means ± SD for all factor combinations\n")

## - Line plots created showing means ± SD for all factor combinations
cat("- Mixed ANOVAs performed for each DV\n")

## - Mixed ANOVAs performed for each DV
cat("- Post-hoc tests for feedback factor (when significant)\n")

## - Post-hoc tests for feedback factor (when significant)
cat("- Hierarchical follow-up ANOVAs for significant interactions\n")

## - Hierarchical follow-up ANOVAs for significant interactions
cat("\nRefer to the sections above for detailed results.")

##
## Refer to the sections above for detailed results.

```

Session Info

```

sessionInfo()

## R version 4.3.3 (2024-02-29)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 24.04.2 LTS
##
## Matrix products: default
## BLAS: /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.12.0

```

```

## LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.12.0
##
## locale:
## [1] LC_CTYPE=en_US.UTF-8      LC_NUMERIC=C
## [3] LC_TIME=de_DE.UTF-8      LC_COLLATE=en_US.UTF-8
## [5] LC_MONETARY=de_DE.UTF-8  LC_MESSAGES=en_US.UTF-8
## [7] LC_PAPER=de_DE.UTF-8     LC_NAME=C
## [9] LC_ADDRESS=C             LC_TELEPHONE=C
## [11] LC_MEASUREMENT=de_DE.UTF-8 LC_IDENTIFICATION=C
##
## time zone: Europe/Berlin
## tzcode source: system (glibc)
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods   base
##
## other attached packages:
## [1] patchwork_1.3.0 knitr_1.50      emmeans_1.11.1 ez_4.4-0
## [5] lubridate_1.9.4 forcats_1.0.0   stringr_1.5.1  dplyr_1.1.4
## [9] purrr_1.0.4     readr_2.1.5     tidyr_1.3.1    tibble_3.2.1
## [13] ggplot2_3.5.1   tidyverse_2.0.0
##
## loaded via a namespace (and not attached):
## [1] gtable_0.3.6      xfun_0.52        lattice_0.22-5   tzdb_0.5.0
## [5] vctrs_0.6.5       tools_4.3.3      Rdpack_2.6.3     generics_0.1.3
## [9] pkgconfig_2.0.3   Matrix_1.6-5     lifecycle_1.0.4  compiler_4.3.3
## [13] farver_2.1.2      munsell_0.5.1    tinytex_0.57     carData_3.0-5
## [17] htmltools_0.5.8.1 yaml_2.3.10      Formula_1.2-5    pillar_1.10.1
## [21] car_3.1-3         nloptr_2.2.1     MASS_7.3-60.0.1  reformulas_0.4.0
## [25] boot_1.3-30       abind_1.4-8      nlme_3.1-164     tidyselect_1.2.1
## [29] digest_0.6.37     mvtnorm_1.3-3    stringi_1.8.4    reshape2_1.4.4
## [33] labeling_0.4.3    splines_4.3.3    fastmap_1.2.0    grid_4.3.3
## [37] colorspace_2.1-1  cli_3.6.4        magrittr_2.0.3    withr_3.0.2
## [41] scales_1.3.0      timechange_0.3.0 estimability_1.5.1 rmarkdown_2.29
## [45] lme4_1.1-36       hms_1.1.3        evaluate_1.0.3    rbibutils_2.3
## [49] mgcv_1.9-1        rlang_1.1.5      Rcpp_1.0.14       glue_1.8.0
## [53] rstudioapi_0.17.1 minqa_1.2.8      R6_2.6.1          plyr_1.8.9

```