MCP ANALYTICS

Group Comparison — t-test, ANOVA & Nonparametric

Statistical analysis report with 9 sections — interactive charts, validated methodology, and AI-generated insights.

Generated **July 18, 2026**

Sections **9**

Platform **mcpanalytics.ai**

Contents

EXECUTIVE SUMMARY

Executive Summary

Key Metrics

Observations	**300**
Groups Compared	**3**
Primary Test	**Welch ANOVA**
Primary p-value	**1.47e-08**
Effect Size	**0.113**
Significant Pairs	**2**

Suggested Interpretation

Yes, weekly sales genuinely differ by store region (Welch ANOVA, p = 1.47e-08). Region B achieves the highest mean (56.392), while Region C records the lowest (50.252). The most pronounced gap is Region B versus Region C: a difference of 6.141 units in favor of Region B (95% CI 3.47 to 8.811, adjusted p = 3.77e-07). Two of three pairwise comparisons are significant; Region A and Region C do not differ meaningfully (difference -0.215, adjusted p = 0.98). The effect size (eta-squared = 0.113) indicates a medium practical magnitude—store region accounts for approximately 11.3% of variation in weekly sales.

OVERVIEW

Analysis Overview

Comparison of Weekly Sales across 3 Store Region groups (300 observations).

Configuration

N Observations	300
N Groups	3
N Tests	4
N Pairwise	3

Suggested Interpretation

This analysis employs a multi-method approach to test whether weekly sales differ across three store regions (A, B, C) using 300 observations. The toolkit combines parametric tests (one-way ANOVA and Welch ANOVA) to compare group means with the rank-based Kruskal-Wallis test, which makes no normality assumption. Effect size (eta-squared) quantifies practical magnitude, while Tukey HSD identifies which specific region pairs differ. The Shapiro-Wilk normality check on residuals (p = 0.388) supports the parametric result as the primary headline, since no strong departure from normality is detected. This dual-test strategy ensures robustness: if parametric and nonparametric verdicts diverge, the safer conclusion emerges.

DATA PREPARATION

Data Quality

Row and group cleaning applied before testing.

Data Quality

Initial Rows	300
Final Rows	300
Rows Removed	0
Groups Dropped	0
Levels Lumped	0

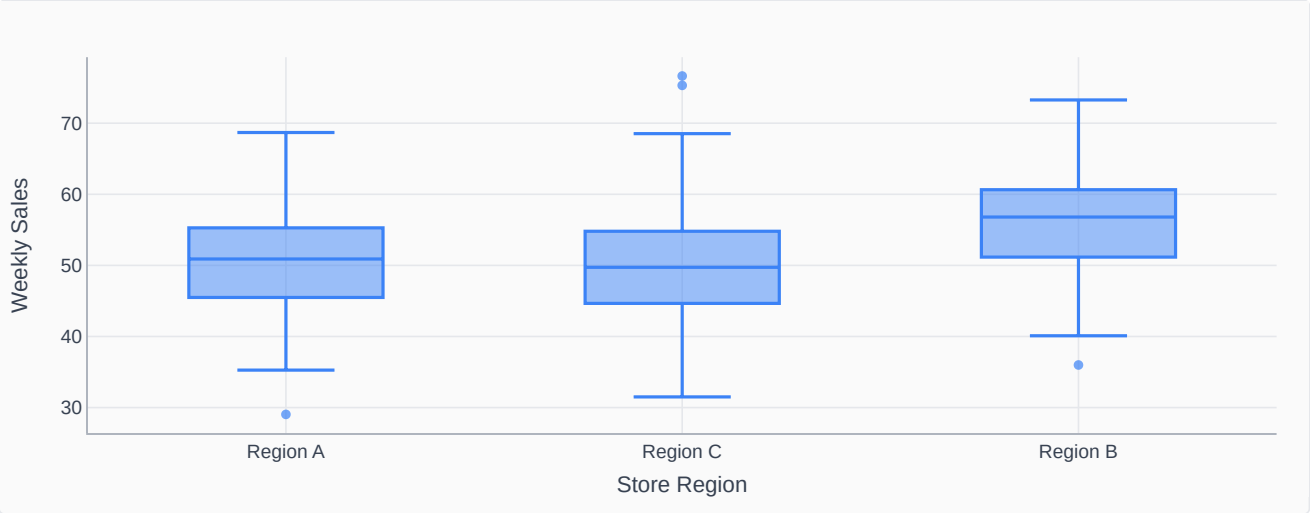
Suggested Interpretation

All 300 rows were retained with no missing values in weekly sales, and no groups were dropped. The three store regions (Region A, Region B, Region C) each contained well above the minimum 3 observations required for analysis. No rare group levels required lumping into an "Other" category. Data quality was complete: the final analysis covers the full 300 observations across all three regions without row loss or group consolidation, establishing a clean foundation for the statistical comparisons that follow.

FIGURE 4

Outcome by Group

Weekly Sales distribution within each Store Region group.



Suggested Interpretation

The boxplot reveals clear separation between Region B and the other two regions. Region B's median (56.8) sits substantially above Region A (50.9) and Region C (49.73). Within-group spreads are broadly comparable, with no evidence of severe heterogeneity that would invalidate the Welch ANOVA. Region A and Region C show substantial overlap in their distributions, consistent with their non-significant pairwise difference (adjusted p = 0.98). The visual separation between Region B's box and those of Regions A and C aligns with the statistical finding of two significant pairwise comparisons, confirming that regional differences in weekly sales are both statistically detectable and visually apparent in the data.

TABLE 5

Group Statistics

n, mean, spread, median and 95% CI of Weekly Sales per Store Region group.

Group	N	Mean	SD	Median	CI Low	CI High
Region A	100	50.47	7.053	50.9	49.07	51.87
Region B	100	56.39	7.726	56.8	54.86	57.92
Region C	100	50.25	9.131	49.73	48.44	52.06

Suggested Interpretation

Region B leads with mean weekly sales of 56.392 (95% CI 54.859 to 57.925, n = 100), followed by Region A at 50.466 (95% CI 49.067 to 51.865, n = 100), and Region C at 50.252 (95% CI 48.44 to 52.063, n = 100). Medians align closely with means within each group (Region B median 56.8, Region A 50.9, Region C 49.73), indicating symmetric distributions. Standard deviations are similar across groups (Region A 7.053, Region B 7.726, Region C 9.131), though Region C shows slightly more variability. The 95% confidence intervals for Region B do not overlap with those of Regions A and C, providing visual confirmation of the significant differences detected in pairwise testing.

TABLE 6

Statistical Tests

The full test battery: parametric, nonparametric, and the normality check.

Test	Statistic	P Value	Effect Size	Interpretation
One-way ANOVA	18.9	1.88e-08	0.113	Mean Weekly Sales differs across Store Region groups: highly significant (p < 0.001); eta-squared = 0.113 (medium effect).
Welch ANOVA (unequal variances)	19.8	1.47e-08	0.113	Variance-robust cross-check: highly significant (p < 0.001).
Kruskal-Wallis	36.72	1.06e-08	—	Rank-based (no normality assumption): highly significant (p < 0.001).
Shapiro-Wilk normality (residuals)	—	0.388	—	No strong evidence against normality.

Suggested Interpretation

All four tests confirm that weekly sales differ significantly across store regions. The Welch ANOVA (recommended headline, p = 1.47e-08, statistic = 19.805) shows the strongest evidence, with eta-squared = 0.113 indicating medium effect size. The standard one-way ANOVA (p = 1.88e-08, statistic = 18.898) and rank-based Kruskal-Wallis (p = 1.06e-08, statistic = 36.716) both corroborate this finding, ensuring robustness across assumptions. The Shapiro-Wilk test on residuals (p = 0.388) shows no strong departure from normality, validating the use of parametric tests. Agreement across all three primary tests—parametric, variance-robust, and nonparametric—provides high confidence that regional differences in weekly sales are genuine.

TABLE 7

Which Groups Differ

Pairwise differences in mean Weekly Sales between Store Region groups.

Comparison	Difference	CI Low	CI High	Adj P	Significant
Region C-Region B	-6.141	-8.811	-3.47	3.77e-07	yes
Region B-Region A	5.926	3.256	8.597	9.73e-07	yes
Region C-Region A	-0.215	-2.885	2.456	0.98	no

Suggested Interpretation

Of three pairwise comparisons, two are statistically significant. The largest and most significant difference is Region B versus Region C: Region B exceeds Region C by 6.141 units (95% CI 3.47 to 8.811, adjusted p = 3.77e-07). Region B also exceeds Region A by 5.926 units (95% CI 3.256 to 8.597, adjusted p = 9.73e-07). In contrast, Region A and Region C show no meaningful difference (-0.215, 95% CI -2.885 to 2.456, adjusted p = 0.98), with the confidence interval including zero. These Tukey HSD estimates are adjusted for multiple comparisons, so the "yes" and "no" verdicts can be read directly without further correction. Region B stands apart as the highest-performing region for weekly sales.

METHODOLOGY

Methodology

Statistical method, assumptions, and diagnostics

Group Comparison — t-test, ANOVA & Nonparametric

Standard-library analysis: does a numeric outcome differ between groups? One deck runs the whole toolkit — Welch and Student t-tests for two groups, one-way and Welch ANOVA for three or more, plus Mann-Whitney and Kruskal-Wallis nonparametric checks, effect sizes (Cohen's d, eta-squared), and pairwise differences with confidence intervals — and tells you which result to trust for your data.

DATA

N = 300 observations

KEY RESULTS

Observations	300
Groups Compared	3
Primary Test	Welch ANOVA
Primary P-Value	1.47e-08
Effect Size	0.1130
Significant Pairs	2

ASSUMPTIONS

- The outcome is numeric (or cleanly convertible) and the group column is categorical
- Observations are independent between and within groups
- Parametric results (t-test/ANOVA) assume roughly normal outcomes per group — the nonparametric tests are reported as a cross-check

LIMITATIONS

- Groups with fewer than 3 observations are excluded from the comparison
- Group columns with many levels are lumped beyond the 8 largest into an Other bucket
- A significant p-value says the groups differ, not why — confounding is not addressed
- Very small groups have little power; a non-significant result is not proof of no difference

SOFTWARE & CITATION

APPENDIX

Analysis Code

Complete R source code for this analysis

Group Comparison — t-test, ANOVA & Nonparametric

Does a numeric outcome differ between groups? One deck runs the whole standard toolkit: Welch and Student t-tests (2 groups) or one-way and Welch ANOVA (3+ groups), Mann-Whitney / Kruskal-Wallis nonparametric cross-checks, effect sizes (Cohen's d, eta-squared), and pairwise differences with confidence intervals.

A.1 Why This Method?

Real questions rarely announce which test they need. Running the parametric and nonparametric versions together, with a normality check on the residuals, lets the report itself say which result is the safest headline — instead of leaving the user to guess between four calculators.

A.2 What This Analysis Covers

- Group distributions side by side (boxplot) and per-group statistics
- The full test battery with statistics, p-values, and effect sizes
- Exactly which group pairs differ (Tukey HSD / Welch CI)

A.3 Standard Library

Platform standard-library module (LAT-1441): runs on ANY dataset via the semantic mapping {outcome, group}. All narrative is derived from the user's own column names and computed values.

```
suppressPackageStartupMessages(library(DT))
suppressPackageStartupMessages(library(htmlwidgets))
suppressPackageStartupMessages(library(arrow))
suppressPackageStartupMessages(library(knitr))
suppressPackageStartupMessages(library(rmarkdown))
suppressPackageStartupMessages(library(dplyr))
suppressPackageStartupMessages(library(tidyr))
suppressPackageStartupMessages(library(ggplot2))
suppressPackageStartupMessages(library(stringr))
suppressPackageStartupMessages(library(lubridate))
suppressPackageStartupMessages(library(broom))
suppressPackageStartupMessages(library(Matrix))
suppressPackageStartupMessages(library(cluster))
suppressPackageStartupMessages(library(data.table))
```

A.4 Core Analysis Pipeline

```
compute_shared <- function(df, params, col_map = list()) {
  # === SHARED EXPORTS ===
  #   initial_rows/final_rows/rows_removed  $ row accounting
  #   outcome_h / group_h  $ humanized user names for the two mapped columns
  #   k / group_levels     $ number of groups after cleaning + their names
  #   n_na_outcome         $ rows dropped for missing/non-numeric outcome
  #   dropped_groups_df    $ data.frame(group, n) — groups dropped (n < 3)
  #   lumped_levels        $ character — levels folded into "Other"
  #   group_summary_df     $ group, n, mean, sd, median, ci_low, ci_high
  #   test_results_df      $ test, statistic, p_value, effect_size, interpretation
  #   pairwise_df          $ comparison, difference, ci_low, ci_high, adj_p, significant
  #   boxplot_df           $ group_name, outcome_value (<= 2000 sampled rows)
  #   shapiro_p / out_frac / nonparam_preferred  $ assumption diagnostics
  #   primary_test / primary_p / primary_sig     $ the headline result
  #   effect_value / effect_text                 $ headline effect size
  #   top_sig_pair         $ 1-row slice of pairwise_df or NULL
  #   metrics / json_output
  # === /SHARED EXPORTS ===
```

Step 1: Resolve mapped columns (humanized for all prose)

```
initial_rows <- nrow(df)
  outcome_h <- humanize_semantic("outcome", col_map)
  group_h   <- humanize_semantic("group", col_map)
  if (!("outcome" %in% names(df)) || !("group" %in% names(df))) {
    stop(sprintf("Group comparison needs both &#x27;%s' (the numeric outcome) and '%s' (the groups) mapped.",
                 outcome_h, group_h))
  }
```

Step 2: Coerce the outcome to numeric (95% rule); drop NA-outcome rows

```
v <- df$outcome
  if (!is.numeric(v)) {
    ch <- as.character(v)
    non_blank <- !is.na(ch) & trimws(ch) != ""
    conv <- suppressWarnings(as.numeric(ch))
    if (sum(non_blank) == 0 ||
        sum(!is.na(conv[non_blank])) < 0.95 * sum(non_blank)) {
      stop(sprintf("The outcome column &#x27;%s' does not look numeric — fewer than 95%% of its values parse as numbers. Pick a
                   outcome_h))
    }
    v <- conv
  }
  df$outcome <- v

  g <- as.character(df$group)
  g[is.na(g) | trimws(g) == ""] <- "Missing"

  keep <- !is.na(df$outcome)
  n_na_outcome <- sum(!keep)
  df <- df[keep, , drop = FALSE]
  g  <- g[keep]
  if (nrow(df) == 0) {
    stop(sprintf("No rows with a usable numeric value in &#x27;%s' remained after cleaning.", outcome_h))
  }
```

Step 3: Clean the groups — drop n<3 (reported), lump beyond 8 levels

```
tab <- table(g)
  small <- names(tab)[tab < 3]
  dropped_groups_df <- data.frame(group = character(0), n = integer(0),
                                  stringsAsFactors = FALSE)
  if (length(small) > 0) {
    dropped_groups_df <- data.frame(group = small, n = as.integer(tab[small]),
                                    stringsAsFactors = FALSE)
    sel <- !(g %in% small)
    df <- df[sel, , drop = FALSE]
    g  <- g[sel]
  }

  lumped_levels <- character(0)
  tab <- sort(table(g), decreasing = TRUE)
  if (length(tab) > 8) {
    keep_lv <- names(tab)[1:8]
    lumped_levels <- setdiff(names(tab), keep_lv)
    g[g %in% lumped_levels] <- "Other"
  }
```

Re-check after lumping ("Other" itself could be tiny)

```
tab <- table(g)
  small2 <- names(tab)[tab < 3]
  if (length(small2) > 0) {
    dropped_groups_df <- rbind(dropped_groups_df,
                               data.frame(group = small2, n = as.integer(tab[small2]),
                                          stringsAsFactors = FALSE))
    sel <- !(g %in% small2)
    df <- df[sel, , drop = FALSE]
    g  <- g[sel]
  }

  gf <- factor(g)
  k  <- nlevels(gf)
  if (k < 2) {
    stop(sprintf("Group comparison needs at least 2 groups in &#x27;%s' with 3 or more rows each; only %d usable group(s) remai
                 group_h, k, group_h))
  }
  y <- df$outcome
  final_rows <- length(y)
  rows_removed <- initial_rows - final_rows
  if (final_rows < 10) {
    stop(sprintf("Only %d usable rows remained — at least 10 are needed to compare groups.", final_rows))
  }
  if (isTRUE(stats::var(y) == 0)) {
    stop(sprintf("The outcome &#x27;%s' has no variation at all (every value is identical) — there is nothing to compare.", outc
  }
  group_levels <- levels(gf)
```

Step 4: Per-group summary statistics with 95% CIs

```
group_summary_df <- do.call(rbind, lapply(group_levels, function(l) {
    x <- y[gf == l]
    n <- length(x)
    m <- mean(x)
    s <- stats::sd(x)
    half <- if (!is.na(s) && s > 0 && n > 1) stats::qt(0.975, n - 1) * s / sqrt(n) else 0
    data.frame(group = l, n = n,
               mean = round(m, 3),
               sd = round(if (is.na(s)) 0 else s, 3),
               median = round(stats::median(x), 3),
               ci_low = round(m - half, 3),
               ci_high = round(m + half, 3),
               stringsAsFactors = FALSE)
  }))
  rownames(group_summary_df) <- NULL
```

Step 5: Assumption diagnostics — normality of residuals + outliers

```
dat <- data.frame(y = y, gf = gf)
  fit <- stats::aov(y ~ gf, data = dat)
  res <- stats::residuals(fit)
  shapiro_p <- NA_real_
  if (final_rows >= 3 && final_rows <= 5000) {
    shapiro_p <- tryCatch(stats::shapiro.test(res)$p.value,
                          error = function(e) NA_real_)
  }
  iqr <- stats::IQR(res)
  qs  <- stats::quantile(res, c(0.25, 0.75), names = FALSE)
  out_frac <- if (is.na(iqr) || iqr == 0) 0 else
    mean(res < qs[1] - 3 * iqr | res > qs[2] + 3 * iqr)
  heavy_outliers <- out_frac > 0.01
  nonparam_preferred <- (!is.na(shapiro_p) && shapiro_p < 0.01) || heavy_outliers

  sig_phrase <- function(p) {
    if (is.na(p)) "could not be computed"
    else if (p < 0.001) "highly significant(p < 0.001)"
    else if (p < 0.05) sprintf("significant(p = %.3g)", p)
    else sprintf("not significant(p = %.3g)", p)
  }
  d_word <- function(d) {
    a <- abs(d)
    if (is.na(a)) "unknown"
    else if (a < 0.2) "negligible" else if (a < 0.5) "small"
    else if (a < 0.8) "medium" else "large"
  }
  eta_word <- function(e) {
    if (is.na(e)) "unknown"
    else if (e < 0.01) "negligible" else if (e < 0.06) "small"
    else if (e < 0.14) "medium" else "large"
  }
```

Step 6: The test battery

```
  test_rows <- list()
    add_test <- function(test, statistic, p_value, effect_size, interpretation) {
      test_rows[[length(test_rows) + 1]] <<- data.frame(
        test = test,
        statistic = round(statistic, 3),
        p_value = signif(p_value, 3),
        effect_size = if (is.na(effect_size)) NA_real_ else round(effect_size, 3),
        interpretation = interpretation,
        stringsAsFactors = FALSE
      )
    }

    pairwise_df <- NULL
    effect_value <- NA_real_
    effect_text <- ""
    primary_test <- NA_character_
    primary_p <- NA_real_

    if (k == 2) {
```

TWO GROUPS: WELCH (PRIMARY), STUDENT, MANN-WHITNEY, COHEN'S D

```
    l1 <- group_levels[1]; l2 <- group_levels[2]
        x1 <- y[gf == l1]; x2 <- y[gf == l2]
        n1 <- length(x1); n2 <- length(x2)
        s1 <- stats::sd(x1); s2 <- stats::sd(x2)
        sp <- sqrt(((n1 - 1) * s1^2 + (n2 - 1) * s2^2) / (n1 + n2 - 2))
        cohens_d <- if (!is.na(sp) && sp > 0) (mean(x1) - mean(x2)) / sp else NA_real_
        higher <- if (mean(x1) >= mean(x2)) l1 else l2

        welch <- tryCatch(stats::t.test(x1, x2), error = function(e) NULL)
        student <- tryCatch(stats::t.test(x1, x2, var.equal = TRUE),
                            error = function(e) NULL)
        mw <- tryCatch(suppressWarnings(stats::wilcox.test(x1, x2)),
                       error = function(e) NULL)
        rank_biserial <- if (!is.null(mw)) 2 * unname(mw$statistic) / (n1 * n2) - 1 else NA_real_

        if (!is.null(welch)) {
          add_test("Welch t-test(unequal variances)", unname(welch$statistic),
                   welch$p.value, cohens_d,
                   sprintf("Difference in mean %s is %s; %s effect(Cohen&#x27;s d), %s mean is higher.",
                           outcome_h, sig_phrase(welch$p.value), d_word(cohens_d), higher))
        }
        if (!is.null(student)) {
          add_test("Student t-test(equal variances)", unname(student$statistic),
                   student$p.value, cohens_d,
                   sprintf("Classic t-test cross-check: %s.", sig_phrase(student$p.value)))
        }
        if (!is.null(mw)) {
          add_test("Mann-Whitney U", unname(mw$statistic), mw$p.value, rank_biserial,
                   sprintf("Rank-based(no normality assumption): %s; rank-biserial r = %s.",
                           sig_phrase(mw$p.value),
                           ifelse(is.na(rank_biserial), "NA", round(rank_biserial, 3))))
        }
        if (!is.na(shapiro_p)) {
          add_test("Shapiro-Wilk normality(residuals)", NA_real_, shapiro_p, NA_real_,
                   if (shapiro_p < 0.01)
                     "Normality clearly violated — favor the rank-based result."
                   else "No strong evidence against normality.")
        }
```

Pairwise table = the single Welch comparison row (ALWAYS has data)

```
    dmean <- mean(x1) - mean(x2)
        if (!is.null(welch)) {
          ci <- as.numeric(welch$conf.int)
          pw_p <- welch$p.value
        } else if (!is.null(student)) {
          ci <- as.numeric(student$conf.int)
          pw_p <- student$p.value
        } else {
          ci <- c(dmean, dmean)
          pw_p <- NA_real_
        }
        pairwise_df <- data.frame(
          comparison = paste0(l1, " - ", l2),
          difference = round(dmean, 3),
          ci_low = round(ci[1], 3),
          ci_high = round(ci[2], 3),
          adj_p = signif(pw_p, 3),
          significant = ifelse(is.na(pw_p), "", ifelse(pw_p < 0.05, "yes", "no")),
          stringsAsFactors = FALSE
        )

        effect_value <- cohens_d
        effect_text <- sprintf("Cohen&#x27;s d = %s (%s effect)",
                               ifelse(is.na(cohens_d), "NA", round(cohens_d, 2)),
                               d_word(cohens_d))
        if (nonparam_preferred && !is.null(mw)) {
          primary_test <- "Mann-Whitney U"; primary_p <- mw$p.value
        } else if (!is.null(welch)) {
          primary_test <- "Welch t-test"; primary_p <- welch$p.value
        } else if (!is.null(student)) {
          primary_test <- "Student t-test"; primary_p <- student$p.value
        } else if (!is.null(mw)) {
          primary_test <- "Mann-Whitney U"; primary_p <- mw$p.value
        } else {
          stop(sprintf("No comparison test could be computed for &#x27;%s' between the two %s groups (the data may be essentially co
                       outcome_h, group_h))
        }
      } else {
```

3+ GROUPS: ANOVA, WELCH ANOVA, KRUSKAL-WALLIS, ETA², TUKEY HSD

```
a_tab <- summary(fit)[[1]]
    ss_b <- a_tab[["Sum Sq"]][1]
    ss_w <- a_tab[["Sum Sq"]][2]
    eta2 <- if (!is.na(ss_b) && !is.na(ss_w) && (ss_b + ss_w) > 0)
      ss_b / (ss_b + ss_w) else NA_real_
    f_stat <- a_tab[["F value"]][1]
    anova_p <- a_tab[["Pr(>F)"]][1]

    welch_a <- tryCatch(stats::oneway.test(y ~ gf, data = dat),
                        error = function(e) NULL)
    kw <- tryCatch(stats::kruskal.test(y ~ gf, data = dat),
                   error = function(e) NULL)

    if (!is.na(anova_p)) {
      add_test("One-way ANOVA", f_stat, anova_p, eta2,
               sprintf("Mean %s differs across %s groups: %s; eta-squared = %s(%s effect).",
                       outcome_h, group_h, sig_phrase(anova_p),
                       ifelse(is.na(eta2), "NA", round(eta2, 3)), eta_word(eta2)))
    }
    if (!is.null(welch_a)) {
      add_test("Welch ANOVA(unequal variances)", unname(welch_a$statistic),
               welch_a$p.value, eta2,
               sprintf("Variance-robust cross-check: %s.", sig_phrase(welch_a$p.value)))
    }
    if (!is.null(kw)) {
      add_test("Kruskal-Wallis", unname(kw$statistic), kw$p.value, NA_real_,
               sprintf("Rank-based(no normality assumption): %s.", sig_phrase(kw$p.value)))
    }
    if (!is.na(shapiro_p)) {
      add_test("Shapiro-Wilk normality(residuals)", NA_real_, shapiro_p, NA_real_,
               if (shapiro_p < 0.01)
                 "Normality clearly violated — favor the rank-based result."
               else "No strong evidence against normality.")
    }

    tk <- tryCatch(stats::TukeyHSD(fit)$gf, error = function(e) NULL)
    if (!is.null(tk) && nrow(tk) > 0) {
      pairwise_df <- data.frame(
        comparison = rownames(tk),
        difference = round(as.numeric(tk[, "diff"]), 3),
        ci_low = round(as.numeric(tk[, "lwr"]), 3),
        ci_high = round(as.numeric(tk[, "upr"]), 3),
        adj_p = signif(as.numeric(tk[, "p adj"]), 3),
        stringsAsFactors = FALSE
      )
      pairwise_df$significant <- ifelse(is.na(pairwise_df$adj_p), "",
                                        ifelse(pairwise_df$adj_p < 0.05, "yes", "no"))
      ok <- !is.na(pairwise_df$difference)
      pairwise_df <- pairwise_df[ok, , drop = FALSE]
      pairwise_df <- pairwise_df[order(-abs(pairwise_df$difference)), , drop = FALSE]
      pairwise_df <- head(pairwise_df, 15)
      rownames(pairwise_df) <- NULL
    } else {
      pairwise_df <- data.frame(comparison = character(0), difference = numeric(0),
                                ci_low = numeric(0), ci_high = numeric(0),
```

```
                                adj_p = numeric(0), significant = character(0),
                                stringsAsFactors = FALSE)
  }

  effect_value <- eta2
  effect_text <- sprintf("eta-squared = %s(%s effect — share of variation in %s explained by %s)",
                         ifelse(is.na(eta2), "NA", round(eta2, 3)),
                         eta_word(eta2), outcome_h, group_h)
  if (nonparam_preferred && !is.null(kw)) {
    primary_test <- "Kruskal-Wallis"; primary_p <- kw$p.value
  } else if (!is.null(welch_a) && !is.na(welch_a$p.value)) {
    primary_test <- "Welch ANOVA"; primary_p <- welch_a$p.value
  } else if (!is.na(anova_p)) {
    primary_test <- "One-way ANOVA"; primary_p <- anova_p
  } else if (!is.null(kw)) {
    primary_test <- "Kruskal-Wallis"; primary_p <- kw$p.value
  } else {
    stop(sprintf("No comparison test could be computed for &#x27;%s' across the %s groups (the data may be essentially constar
                 outcome_h, group_h))
  }
}

test_results_df <- do.call(rbind, test_rows)
rownames(test_results_df) <- NULL
primary_sig <- !is.na(primary_p) && primary_p < 0.05
```

Top significant pairwise difference (NA-safe — never which.max over NAs)

```
top_sig_pair <- NULL
if (nrow(pairwise_df) > 0) {
  sig_rows <- pairwise_df[!is.na(pairwise_df$difference) &
                          pairwise_df$significant == "yes", , drop = FALSE]
  if (nrow(sig_rows) > 0) top_sig_pair <- sig_rows[1, ]  # already |diff|-sorted
}
n_sig_pairs <- sum(pairwise_df$significant == "yes", na.rm = TRUE)
```

Step 7: Boxplot dataset — <= 2000 sampled rows

```
  set.seed(42)
    sidx <- if (final_rows > 2000) sample(final_rows, 2000) else seq_len(final_rows)
    boxplot_df <- data.frame(
      group_name = as.character(gf[sidx]),
      outcome_value = y[sidx],
      stringsAsFactors = FALSE
    )

    metrics <- list(
      `Observations`      = final_rows,
      `Groups Compared`   = k,
      `Primary Test`      = primary_test,
      `Primary p-value`   = signif(primary_p, 3),
      `Effect Size`       = if (is.na(effect_value)) NA_real_ else round(effect_value, 3),
      `Significant Pairs` = as.integer(n_sig_pairs)
    )

    json_output <- list(
      answer = paste0(
        "Compared ", outcome_h, " across ", k, " ", group_h, " groups(",
        format(final_rows, big.mark = ","), " rows): the ", primary_test,
        " finds the difference ", sig_phrase(primary_p), "; ", effect_text, ". ",
        if (!is.null(top_sig_pair)) paste0(
          "Largest significant gap: ", top_sig_pair$comparison, " (difference = ",
          top_sig_pair$difference, ", adjusted p = ", top_sig_pair$adj_p, "); ")
        else "No individual pair reaches significance; ",
        n_sig_pairs, " of ", nrow(pairwise_df), " pairwise comparison(s) significant at p<0.05."
      ),
      cards = lapply(
        c("tldr", "overview", "preprocessing", "group_boxplot",
          "group_summary", "test_results", "pairwise_differences"),
        function(cid) list(id = cid, metrics = metrics)
      )
    )

    list(
      initial_rows = initial_rows, final_rows = final_rows,
      rows_removed = rows_removed, n_na_outcome = n_na_outcome,
      outcome_h = outcome_h, group_h = group_h,
      k = k, group_levels = group_levels,
      dropped_groups_df = dropped_groups_df, lumped_levels = lumped_levels,
      group_summary_df = group_summary_df,
      test_results_df = test_results_df,
      pairwise_df = pairwise_df,
      boxplot_df = boxplot_df,
      shapiro_p = shapiro_p, out_frac = out_frac,
      nonparam_preferred = nonparam_preferred,
      primary_test = primary_test, primary_p = primary_p,
      primary_sig = primary_sig,
      effect_value = effect_value, effect_text = effect_text,
      top_sig_pair = top_sig_pair, n_sig_pairs = n_sig_pairs,
      metrics = metrics, json_output = json_output
    )
  }
```

This is the exact R code executed to produce the analysis above. Run it with the same dataset to reproduce these results.