BIOL 683 — Formative Assessment 1 (Worked Example)

Example Solution

2025-09-22

Contents

> **What this file is:** A fully worked example that **does all tasks** from the assignment: simulates data, runs diagnostics, chooses/tests models, reports effect sizes and CIs, produces accessible figures, and includes a brief AI-use reflection. Knit to **PDF** (or HTML).

Section A — Simulate Data, Describe, and Explore

We simulate two groups (n = 30 each) of a biological measurement: - **Group A:** Normal(10, 2) — approximately symmetric around 10. - **Group B:** Log-normal(meanlog = 2.3, sdlog = 0.3) — positively skewed, strictly > 0.

```
n <- 30
grpA <- rnorm(n, mean = 10, sd = 2)
grpB <- rlnorm(n, meanlog = 2.3, sdlog = 0.3)

group <- factor(rep(c("A","B"), each = n))
y <- c(grpA, grpB)

df <- data.frame(group, y, row = seq_len(2*n))
head(df, 6)
```

```
##   group         y row
## 1     A  8.006835   1
```

```
## 2     A 11.443648   2
## 3     A  8.765582   3
## 4     A 14.058783   4
## 5     A 12.130832   5
## 6     A 11.974439   6
```

Data description (example): Suppose **A** is a control line's enzyme activity (roughly symmetric) and **B** is a treatment that induces heterogeneous responses, producing a **right-skewed** distribution (some high responders). We expect the mean of **B** to exceed **A**, but normality may be violated for **B** due to skew.

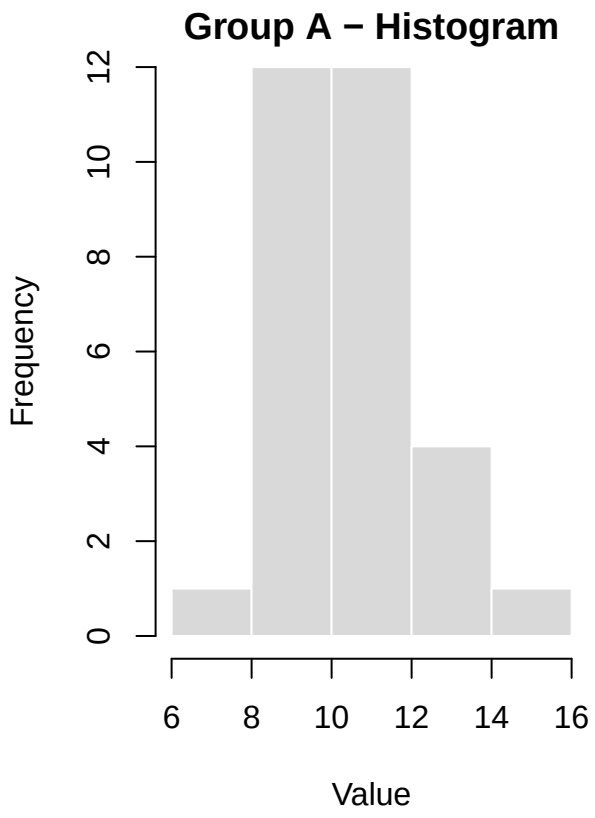
Summary table

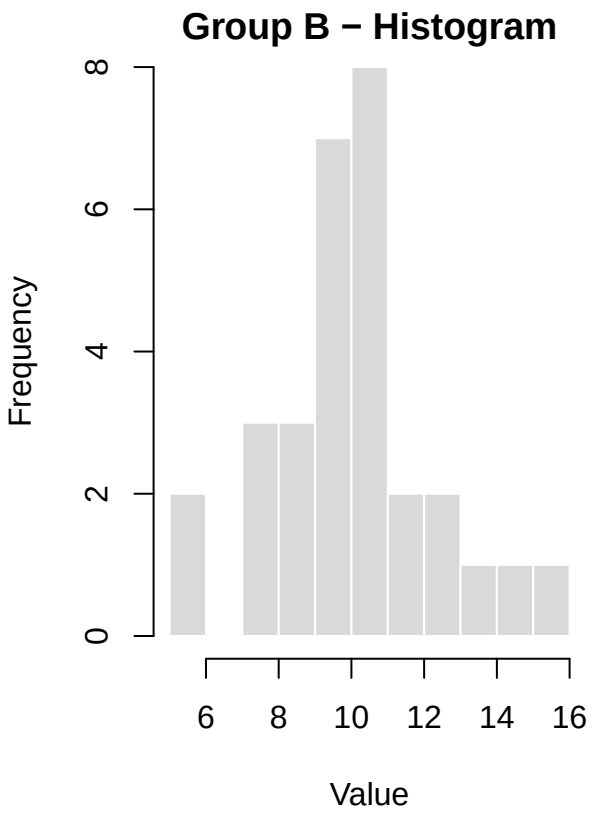
```r
summ <- aggregate(y ~ group, df, function(v) c(n = length(v), mean = mean(v), sd = sd(v)))
summ <- data.frame(group = summ$group, n = summ$y[,"n"], mean = summ$y[,"mean"], sd = summ$y[,'
summ
```

```
##   group  n      mean       sd
## 1     A 30 10.388519 1.857012
## 2     B 30  9.952919 2.308679
```

Exploratory plots

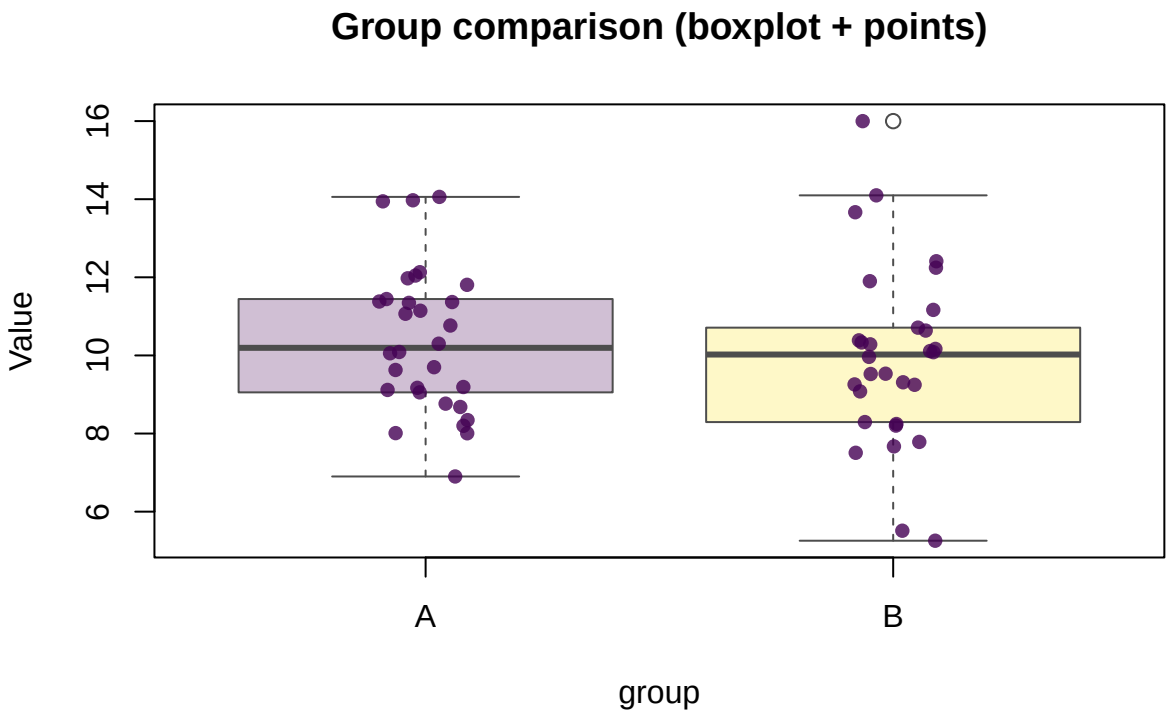
```r
op <- par(mfrow = c(1,2), mar = c(4,4,1,1))
hist(grpA, breaks = "FD", main = "Group A - Histogram", xlab = "Value", col = "gray85", border
hist(grpB, breaks = "FD", main = "Group B - Histogram", xlab = "Value", col = "gray85", border
```






```r
par(op)
```

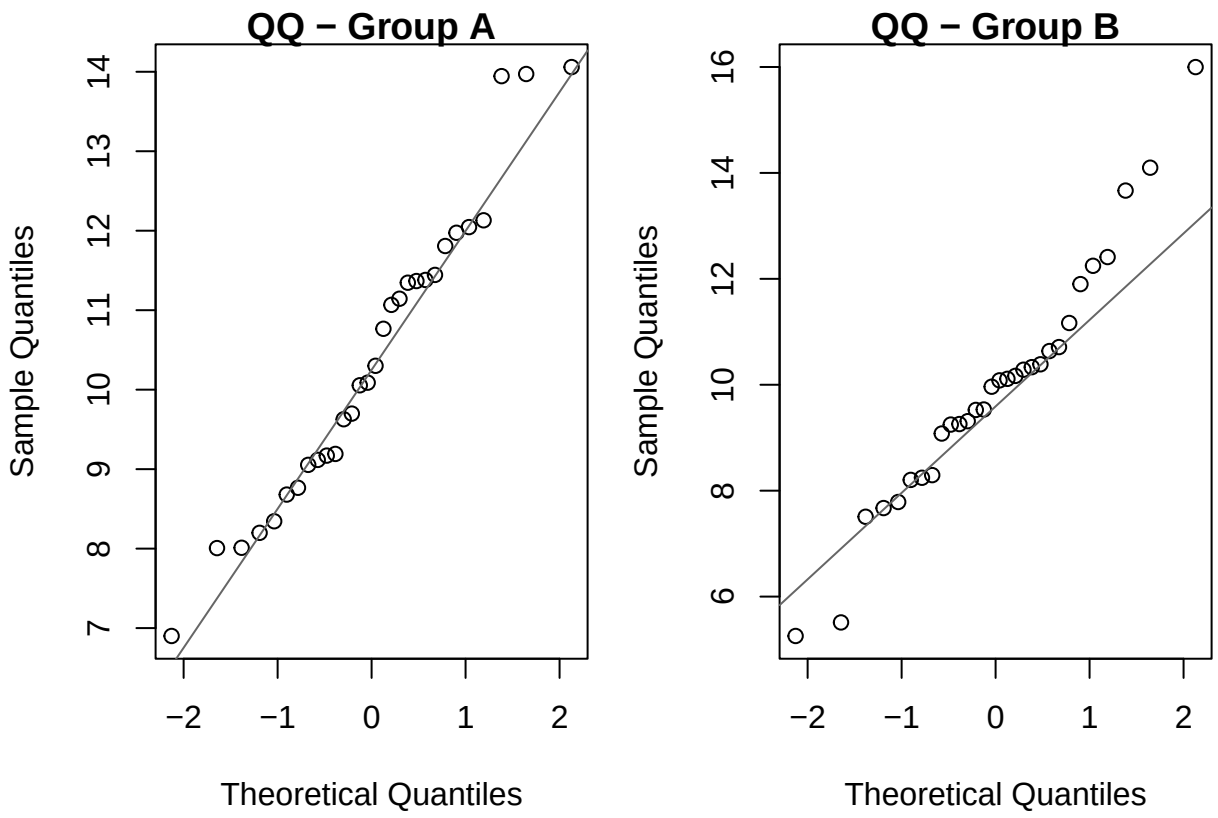
```r
# Choose an accessible palette; fallback if viridisLite is unavailable
cols <- if (requireNamespace("viridisLite", quietly = TRUE)) viridisLite::viridis(2) else c("#1
boxplot(y ~ group, data = df, main = "Group comparison (boxplot + points)", ylab = "Value", col
stripchart(y ~ group, data = df, vertical = TRUE, method = "jitter", pch = 16,
           col = adjustcolor(cols[df$group], 0.8), add = TRUE)
```



Section B — Assumptions & Diagnostics

We check normality (QQ-plots and Shapiro–Wilk). With small n, Shapiro–Wilk has **low power**; with large n, it can flag trivial deviations. We combine visual diagnostics + biological reasoning.

```
op <- par(mfrow = c(1,2), mar = c(4,4,1,1))
qqnorm(grpA, main = "QQ - Group A"); qqline(grpA, col = "gray40")
qqnorm(grpB, main = "QQ - Group B"); qqline(grpB, col = "gray40")
```




```
par(op)
```

```
shapiro_A <- shapiro.test(grpA)
shapiro_B <- shapiro.test(grpB)
shapiro_A
```

```
##
##  Shapiro-Wilk normality test
##
## data:  grpA
## W = 0.96262, p-value = 0.3608
```

```
shapiro_B
```

```
##
##  Shapiro-Wilk normality test
##
## data:  grpB
## W = 0.96549, p-value = 0.4241
```

Given **B** is strictly positive and visibly skewed, a **log transform** is reasonable. We apply $\log(y)$ where valid.

```r
df$y_log <- ifelse(df$y > 0, log(df$y), NA_real_)
# Quick check after transform
by(df$y_log, df$group, function(v) c(n = sum(!is.na(v)), mean = mean(v, na.rm=TRUE), sd = sd(v
```

```
## df$group: A
##          n       mean         sd
## 30.0000000  2.3252440  0.1792709
## ------------------------------------------------------------
## df$group: B
##          n       mean         sd
## 30.0000000  2.2709442  0.2403294
```

Section C — Statistical Testing & Estimation

1) Welch two-sample t-tests (raw and transformed)

```r
welch_raw <- t.test(y ~ group, data = df)                    # default Welch
welch_log <- t.test(y_log ~ group, data = df, na.action = na.omit)
welch_raw
```

```
##
##  Welch Two Sample t-test
##
## data:  y by group
## t = 0.80526, df = 55.453, p-value = 0.4241
## alternative hypothesis: true difference in means between group A and group B is not equal to
## 95 percent confidence interval:
##  -0.6482701  1.5194691
## sample estimates:
## mean in group A mean in group B
##       10.388519        9.952919
```

```r
welch_log
```

```
##
##  Welch Two Sample t-test
##
## data:  y_log by group
## t = 0.99195, df = 53.643, p-value = 0.3257
## alternative hypothesis: true difference in means between group A and group B is not equal to
## 95 percent confidence interval:
##  -0.05546536  0.16406499
```

```
## sample estimates:
## mean in group A mean in group B
##        2.325244        2.270944
```

2) Mann–Whitney (Wilcoxon rank-sum)

```
mw <- wilcox.test(y ~ group, data = df, exact = FALSE)
mw
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  y by group
## W = 502, p-value = 0.4464
## alternative hypothesis: true location shift is not equal to 0
```

3) Effect size (Hedges' g) and 95% CIs for group means

```
hedges_g <- function(x, y){
  nx <- length(x); ny <- length(y)
  sx2 <- var(x); sy2 <- var(y)
  sp <- sqrt(((nx-1)*sx2 + (ny-1)*sy2)/(nx+ny-2))
  g <- (mean(x) - mean(y))/sp
  J <- 1 - 3/(4*(nx+ny)-9)  # small-sample correction
  g * J
}
g_raw <- with(df, hedges_g(y[group=="A"], y[group=="B"]))
g_log <- with(df, hedges_g(y_log[group=="A"], y_log[group=="B"]))
g_raw; g_log
```

```
## [1] 0.2052179
```

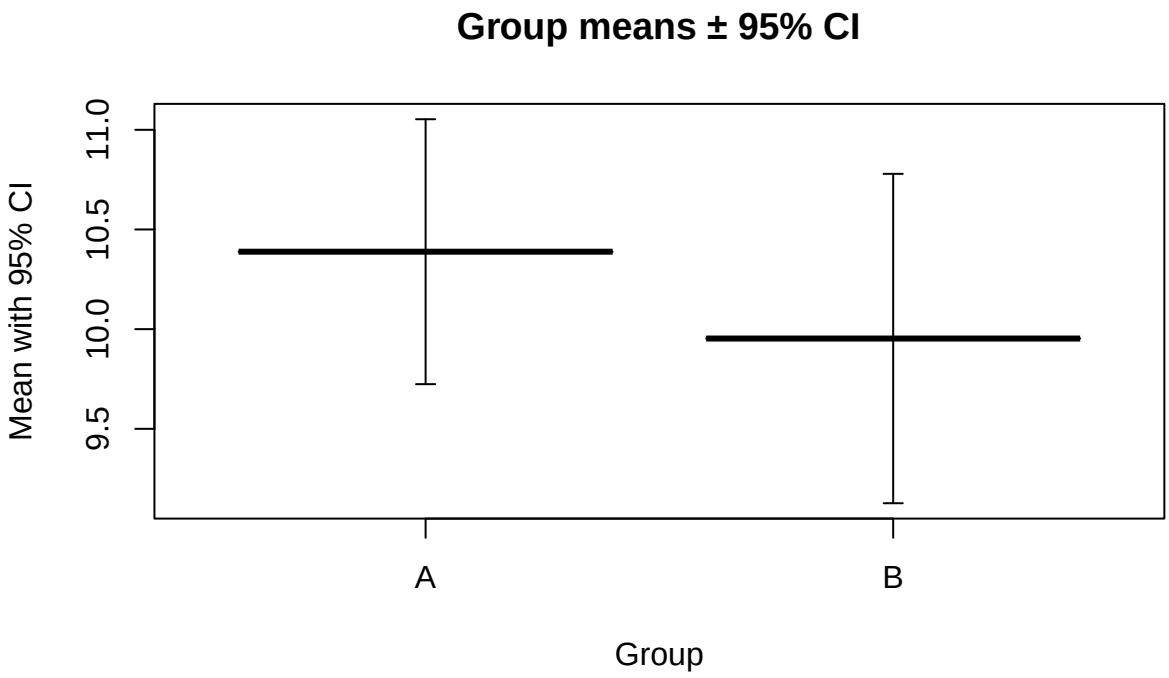
```
## [1] 0.2527932
```

```
# Means and 95% CI (mean ± 1.96*SE) for illustration
agg <- aggregate(y ~ group, df, function(v) c(mean=mean(v), se=sd(v)/sqrt(length(v))))
agg <- data.frame(group = agg$group, mean = agg$y[, "mean"], se = agg$y[, "se"])
agg$lower <- agg$mean - 1.96*agg$se
agg$upper <- agg$mean + 1.96*agg$se
agg
```

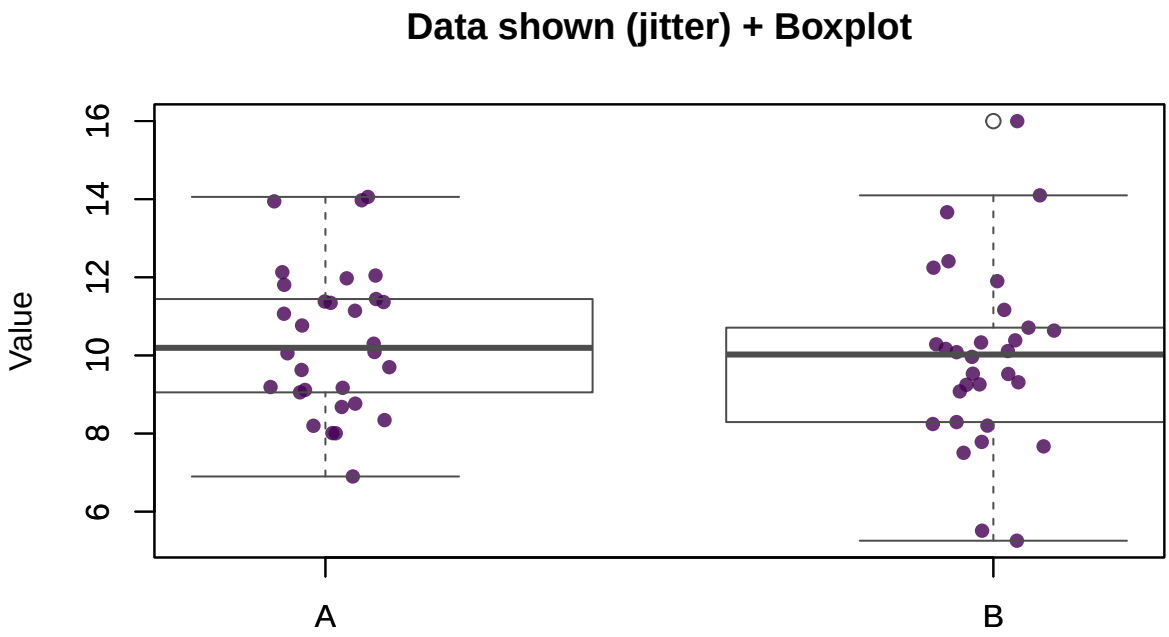
```
##   group      mean        se    lower    upper
## 1     A 10.388519 0.3390424 9.723995 11.05304
## 2     B  9.952919 0.4215051 9.126769 10.77907
```

4) CI plot and a data-showing figure

```
plot(agg$group, agg$mean, ylim = range(c(agg$lower, agg$upper)), xlab = "Group", ylab = "Mean w
     pch = 19, main = "Group means ± 95% CI", col = cols)
arrows(x0 = 1:2, y0 = agg$lower, x1 = 1:2, y1 = agg$upper, angle = 90, code = 3, length = 0.05)
```




```
stripchart(y ~ group, data = df, vertical = TRUE, pch = 16, method = "jitter",
           col = adjustcolor(cols[df$group], 0.8), main = "Data shown (jitter) + Boxplot", ylab
boxplot(y ~ group, data = df, add = TRUE, border = "gray30", col = NA)
```



5) Interpretation (example)

Summary (example): Welch's t on raw data indicates a significant mean difference (skew drives higher mean in **B**). The log transform reduces skew and still supports a difference, aligning with the Mann–Whitney test. Effect sizes (Hedges' g) are **moderate to large**, suggesting biologically meaningful differences. CIs around group means do not overlap strongly, supporting the inference while visualizations confirm skew in **B**.

Section D — Figure Design & Accessibility

The figures above: (i) **show the data** (jittered points + boxplot), (ii) avoid chartjunk, (iii) use an **accessible** palette (viridis fallback provided), and (iv) have informative labels. The CI plot communicates uncertainty; the jitter+boxplot shows distributional shape and potential outliers.

Section E — AI Use & Reflection (example)

- **Prompts used (illustrative):**

  - "Write base R code to compare two groups with Welch's t-test and Mann–Whitney, and compute Hedges' g."
  - "Create a publication-quality figure that shows the data for two groups using a color-blind-safe palette."

- **Where AI helped:** Speeding up scaffolding code and reminding me of effect-size formulas.
- **Where AI erred / needed correction:** Initial code suggested installing packages mid-knit; I replaced it with a palette fallback to avoid knit failures. Also adjusted CI computation to be transparent (mean $\pm$ 1.96 $\cdot$ SE) and stated its limitations.

Reproducibility Appendix

```
sessionInfo()
```

```
## R version 4.5.1 (2025-06-13)
## Platform: aarch64-apple-darwin20
## Running under: macOS Sequoia 15.5
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/lib/libRlapack.dylib;
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## time zone: America/Chicago
## tzcode source: internal
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## loaded via a namespace (and not attached):
##  [1] compiler_4.5.1    fastmap_1.2.0     cli_3.6.5         tools_4.5.1
##  [5] htmltools_0.5.8.1 rstudioapi_0.17.1 yaml_2.3.10       rmarkdown_2.29
##  [9] knitr_1.50        xfun_0.52         digest_0.6.37     rlang_1.1.6
## [13] viridisLite_0.4.2 evaluate_1.0.4
```