bumblebee-bat-enhanced-analysis-with-dark-theme.R

jonathanrocha

2025-01-18

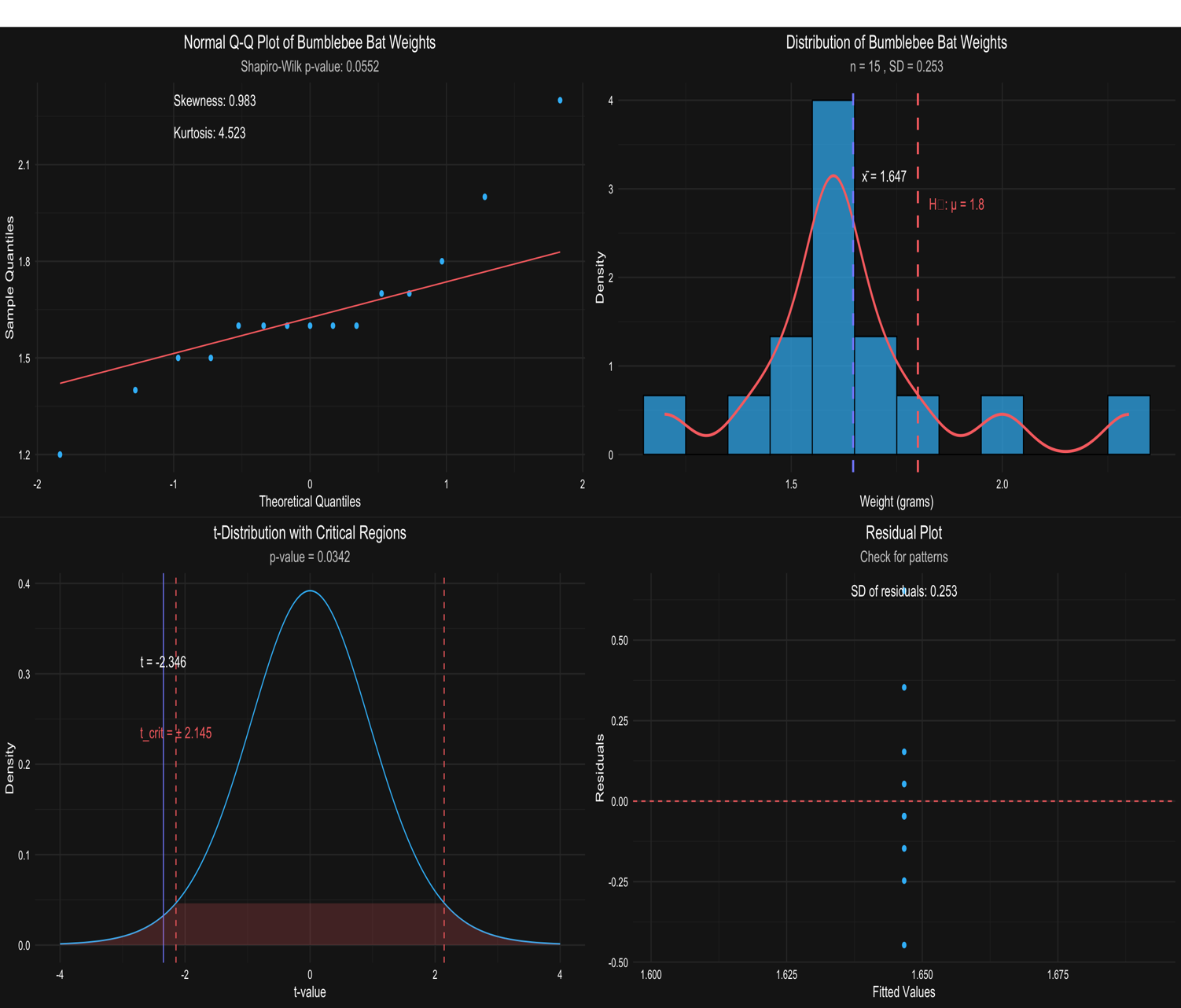
# Load required libraries  
library(ggplot2)  
library(gridExtra)  
library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.4 ✔ readr 2.1.5  
## ✔ forcats 1.0.0 ✔ stringr 1.5.1  
## ✔ lubridate 1.9.4 ✔ tibble 3.2.1  
## ✔ purrr 1.0.2 ✔ tidyr 1.3.1  
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::combine() masks gridExtra::combine()  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(nortest) # For additional normality tests  
library(moments) # For skewness and kurtosis tests  
  
# Set up dark theme for all plots  
dark\_theme <- theme\_minimal() +  
 theme(  
 text = element\_text(color = "white"),  
 axis.text = element\_text(color = "white"),  
 axis.title = element\_text(color = "white"),  
 plot.title = element\_text(color = "white", hjust = 0.5),  
 plot.subtitle = element\_text(color = "#cccccc", hjust = 0.5),  
 plot.background = element\_rect(fill = "#1a1a1a", color = NA),  
 panel.background = element\_rect(fill = "#1a1a1a", color = NA),  
 panel.grid.major = element\_line(color = "#333333"),  
 panel.grid.minor = element\_line(color = "#2b2b2b"),  
 legend.background = element\_rect(fill = "#1a1a1a"),  
 legend.text = element\_text(color = "white"),  
 legend.title = element\_text(color = "white")  
 )  
  
# Create data frame  
bat\_data <- data.frame(  
 weight = c(1.7, 1.6, 1.5, 2.0, 2.3, 1.6, 1.6, 1.8, 1.5, 1.7, 1.2, 1.4, 1.6, 1.6, 1.6)  
)  
  
# Calculate comprehensive summary statistics  
summary\_stats <- data.frame(  
 n = length(bat\_data$weight),  
 mean = mean(bat\_data$weight),  
 median = median(bat\_data$weight),  
 sd = sd(bat\_data$weight),  
 se = sd(bat\_data$weight)/sqrt(length(bat\_data$weight)),  
 skewness = skewness(bat\_data$weight),  
 kurtosis = kurtosis(bat\_data$weight),  
 t\_stat = (mean(bat\_data$weight) - 1.8)/(sd(bat\_data$weight)/sqrt(length(bat\_data$weight)))  
)  
  
# Additional Statistical Tests  
# Shapiro-Wilk test for normality  
sw\_test <- shapiro.test(bat\_data$weight)  
# Anderson-Darling test for normality  
ad\_test <- ad.test(bat\_data$weight)  
# One-sample t-test  
t\_test\_result <- t.test(bat\_data$weight, mu = 1.8)  
# Sign test for median  
sign\_test <- binom.test(  
 sum(bat\_data$weight > 1.8),  
 length(bat\_data$weight),  
 p = 0.5,  
 alternative = "two.sided"  
)  
  
# Create QQ Plot with annotations  
qq\_plot <- ggplot(bat\_data, aes(sample = weight)) +  
 stat\_qq(color = "#38bdf8") +   
 stat\_qq\_line(color = "#f87171") +  
 labs(title = "Normal Q-Q Plot of Bumblebee Bat Weights",  
 subtitle = paste("Shapiro-Wilk p-value:", round(sw\_test$p.value, 4)),  
 x = "Theoretical Quantiles",  
 y = "Sample Quantiles") +  
 annotate("text", x = -1, y = max(bat\_data$weight),  
 label = paste("Skewness:", round(summary\_stats$skewness, 3)),  
 color = "white", hjust = 0) +  
 annotate("text", x = -1, y = max(bat\_data$weight) - 0.1,  
 label = paste("Kurtosis:", round(summary\_stats$kurtosis, 3)),  
 color = "white", hjust = 0) +  
 dark\_theme  
  
# Create Enhanced Histogram with density curve  
hist\_plot <- ggplot(bat\_data, aes(x = weight)) +  
 geom\_histogram(aes(y = ..density..), binwidth = 0.1,   
 fill = "#38bdf8", color = "black", alpha = 0.7) +  
 geom\_density(color = "#f87171", linewidth = 1) +  
 geom\_vline(xintercept = mean(bat\_data$weight),   
 color = "#818cf8", linetype = "dashed", linewidth = 1) +  
 geom\_vline(xintercept = 1.8,   
 color = "#f87171", linetype = "dashed", linewidth = 1) +  
 annotate("text", x = mean(bat\_data$weight), y = max(density(bat\_data$weight)$y),  
 label = paste("x̄ =", round(mean(bat\_data$weight), 3)),  
 color = "white", hjust = -0.2) +  
 annotate("text", x = 1.8, y = max(density(bat\_data$weight)$y) \* 0.9,  
 label = "H₀: μ = 1.8",  
 color = "#f87171", hjust = -0.2) +  
 labs(title = "Distribution of Bumblebee Bat Weights",  
 subtitle = paste("n =", length(bat\_data$weight), ", SD =", round(sd(bat\_data$weight), 3)),  
 x = "Weight (grams)",  
 y = "Density") +  
 dark\_theme  
  
# Create enhanced t-distribution plot  
t\_stat <- summary\_stats$t\_stat  
df <- summary\_stats$n - 1  
t\_crit <- qt(0.975, df)  
  
t\_vals <- seq(-4, 4, length.out = 1000)  
t\_dens <- dt(t\_vals, df)  
t\_dist\_data <- data.frame(t = t\_vals, density = t\_dens)  
  
t\_plot <- ggplot(t\_dist\_data, aes(x = t, y = density)) +  
 geom\_line(color = "#38bdf8") +  
 geom\_vline(xintercept = c(-t\_crit, t\_crit),   
 color = "#f87171", linetype = "dashed") +  
 geom\_vline(xintercept = t\_stat,   
 color = "#818cf8", linetype = "solid") +  
 geom\_ribbon(data = subset(t\_dist\_data, t <= -t\_crit | t >= t\_crit),  
 aes(ymax = density, ymin = 0),   
 fill = "#f87171", alpha = 0.2) +  
 annotate("text", x = t\_stat, y = max(t\_dens) \* 0.8,  
 label = paste("t =", round(t\_stat, 3)),  
 color = "white") +  
 annotate("text", x = -t\_crit, y = max(t\_dens) \* 0.6,  
 label = paste("t\_crit = ±", round(t\_crit, 3)),  
 color = "#f87171") +  
 labs(title = "t-Distribution with Critical Regions",  
 subtitle = paste("p-value =", round(t\_test\_result$p.value, 4)),  
 x = "t-value",  
 y = "Density") +  
 dark\_theme  
  
# Create enhanced residual plot  
residuals <- bat\_data$weight - mean(bat\_data$weight)  
fitted\_values <- rep(mean(bat\_data$weight), length(bat\_data$weight))  
  
residual\_plot <- ggplot(data.frame(fitted = fitted\_values, residuals = residuals),   
 aes(x = fitted, y = residuals)) +  
 geom\_point(color = "#38bdf8") +  
 geom\_hline(yintercept = 0, linetype = "dashed", color = "#f87171") +  
 geom\_smooth(method = "loess", color = "#818cf8", se = TRUE, fill = "#818cf8", alpha = 0.2) +  
 annotate("text", x = mean(fitted\_values), y = max(residuals),  
 label = paste("SD of residuals:", round(sd(residuals), 3)),  
 color = "white") +  
 labs(title = "Residual Plot",  
 subtitle = "Check for patterns",  
 x = "Fitted Values",  
 y = "Residuals") +  
 dark\_theme  
  
# Arrange all plots in a grid with title  
grid.arrange(qq\_plot, hist\_plot, t\_plot, residual\_plot,  
 ncol = 2,  
 top = grid::textGrob("Bumblebee Bat Weight Analysis",   
 gp = grid::gpar(col = "white", fontsize = 16)))

## Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0.  
## ℹ Please use `after\_stat(density)` instead.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.

## `geom\_smooth()` using formula = 'y ~ x'



# Print comprehensive statistical summary  
cat("\nComprehensive Statistical Analysis of Bumblebee Bat Weights\n")

##   
## Comprehensive Statistical Analysis of Bumblebee Bat Weights

cat("=======================================================\n\n")

## =======================================================

cat("Basic Summary Statistics:\n")

## Basic Summary Statistics:

cat("-----------------------\n")

## -----------------------

cat("Sample Size:", summary\_stats$n, "\n")

## Sample Size: 15

cat("Mean:", round(summary\_stats$mean, 3), "grams\n")

## Mean: 1.647 grams

cat("Median:", round(summary\_stats$median, 3), "grams\n")

## Median: 1.6 grams

cat("Standard Deviation:", round(summary\_stats$sd, 3), "grams\n")

## Standard Deviation: 0.253 grams

cat("Standard Error:", round(summary\_stats$se, 3), "grams\n")

## Standard Error: 0.065 grams

cat("Skewness:", round(summary\_stats$skewness, 3), "\n")

## Skewness: 0.983

cat("Kurtosis:", round(summary\_stats$kurtosis, 3), "\n\n")

## Kurtosis: 4.523

cat("Normality Tests:\n")

## Normality Tests:

cat("---------------\n")

## ---------------

cat("Shapiro-Wilk Test:\n")

## Shapiro-Wilk Test:

cat(" W =", round(sw\_test$statistic, 4), "\n")

## W = 0.8844

cat(" p-value =", round(sw\_test$p.value, 4), "\n\n")

## p-value = 0.0552

cat("Anderson-Darling Test:\n")

## Anderson-Darling Test:

cat(" A =", round(ad\_test$statistic, 4), "\n")

## A = 0.8755

cat(" p-value =", round(ad\_test$p.value, 4), "\n\n")

## p-value = 0.0185

cat("One-Sample t-Test Results:\n")

## One-Sample t-Test Results:

cat("------------------------\n")

## ------------------------

cat("H₀: μ = 1.8 grams\n")

## H₀: μ = 1.8 grams

cat("t-statistic:", round(t\_test\_result$statistic, 3), "\n")

## t-statistic: -2.346

cat("degrees of freedom:", t\_test\_result$parameter, "\n")

## degrees of freedom: 14

cat("p-value:", round(t\_test\_result$p.value, 4), "\n")

## p-value: 0.0342

cat("95% CI:", round(t\_test\_result$conf.int[1], 3), "to",   
 round(t\_test\_result$conf.int[2], 3), "grams\n\n")

## 95% CI: 1.506 to 1.787 grams

cat("Sign Test Results:\n")

## Sign Test Results:

cat("----------------\n")

## ----------------

cat("H₀: median = 1.8 grams\n")

## H₀: median = 1.8 grams

cat("Number of values > 1.8:", sign\_test$statistic, "\n")

## Number of values > 1.8: 2

cat("p-value:", round(sign\_test$p.value, 4), "\n")

## p-value: 0.0074