bumblebee-bat-r-viz.r

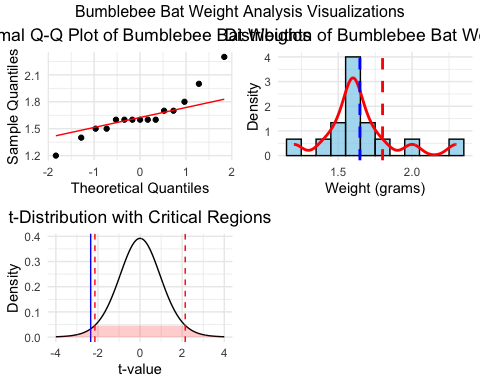
jonathanrocha

2025-01-18

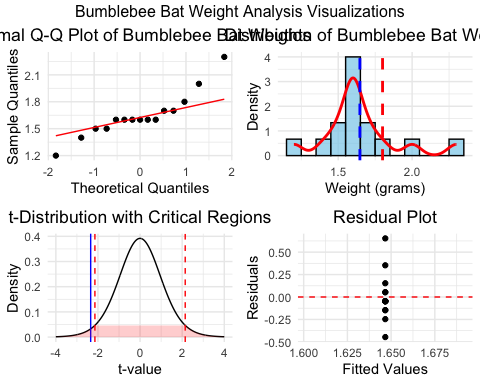
# Load required libraries  
library(ggplot2)  
library(gridExtra)  
library(tidyverse)

## ── 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

# 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)  
)  
  
# 1. Create QQ Plot for normality check  
qq\_plot <- ggplot(bat\_data, aes(sample = weight)) +  
 stat\_qq() +   
 stat\_qq\_line(color = "red") +  
 labs(title = "Normal Q-Q Plot of Bumblebee Bat Weights",  
 x = "Theoretical Quantiles",  
 y = "Sample Quantiles") +  
 theme\_minimal() +  
 theme(plot.title = element\_text(hjust = 0.5))  
  
# 2. Create Histogram with density curve  
hist\_plot <- ggplot(bat\_data, aes(x = weight)) +  
 geom\_histogram(aes(y = after\_stat(density)), binwidth = 0.1,   
 fill = "skyblue", color = "black", alpha = 0.7) +  
 geom\_density(color = "red", linewidth = 1) +  
 geom\_vline(xintercept = mean(bat\_data$weight),   
 color = "blue", linetype = "dashed", linewidth = 1) +  
 geom\_vline(xintercept = 1.8,   
 color = "red", linetype = "dashed", linewidth = 1) +  
 labs(title = "Distribution of Bumblebee Bat Weights",  
 x = "Weight (grams)",  
 y = "Density") +  
 theme\_minimal() +  
 theme(plot.title = element\_text(hjust = 0.5))  
  
# 3. Create t-distribution plot with critical regions  
t\_stat <- (mean(bat\_data$weight) - 1.8)/(sd(bat\_data$weight)/sqrt(length(bat\_data$weight)))  
df <- length(bat\_data$weight) - 1  
t\_crit <- qt(0.975, df)  
  
# Create sequence of t-values  
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() +  
 geom\_vline(xintercept = c(-t\_crit, t\_crit),   
 color = "red", linetype = "dashed") +  
 geom\_vline(xintercept = t\_stat,   
 color = "blue", linetype = "solid") +  
 geom\_ribbon(data = subset(t\_dist\_data, t <= -t\_crit | t >= t\_crit),  
 aes(ymax = density, ymin = 0),   
 fill = "red", alpha = 0.2) +  
 labs(title = "t-Distribution with Critical Regions",  
 x = "t-value",  
 y = "Density") +  
 theme\_minimal() +  
 theme(plot.title = element\_text(hjust = 0.5))  
  
# Arrange all plots in a grid  
grid.arrange(qq\_plot, hist\_plot, t\_plot,   
 ncol = 2,  
 top = "Bumblebee Bat Weight Analysis Visualizations")



# Save the plots if needed  
ggsave("bumblebee\_analysis\_plots.pdf",   
 arrangeGrob(qq\_plot, hist\_plot, t\_plot, ncol = 2),  
 width = 12, height = 8)  
  
# Add residual plot  
t\_test\_result <- t.test(bat\_data$weight, mu = 1.8)  
residuals <- bat\_data$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() +  
 geom\_hline(yintercept = 0, linetype = "dashed", color = "red") +  
 labs(title = "Residual Plot",  
 x = "Fitted Values",  
 y = "Residuals") +  
 theme\_minimal() +  
 theme(plot.title = element\_text(hjust = 0.5))  
  
# Update grid with residual plot  
grid.arrange(qq\_plot, hist\_plot, t\_plot, residual\_plot,  
 ncol = 2,  
 top = "Bumblebee Bat Weight Analysis Visualizations")



# Save updated plots  
ggsave("bumblebee\_analysis\_plots\_with\_residuals.pdf",  
 arrangeGrob(qq\_plot, hist\_plot, t\_plot, residual\_plot, ncol = 2),  
 width = 12, height = 10)