```
# APPENDIX
# Dogs Rehoming Times: A Statistical Comparison between Three Breeds -- Code

# Load libraries
library(MASS)

# ===== 1. LOAD DATA =====
load('dog_rehoming_data.RData')
df <- mysample

cat("\n")
paste(rep("=", 70), collapse="")
cat("DOG REHOMING TIME ANALYSIS\n")
paste(rep("=", 70), collapse="")

# ===== 2. DATA INSPECTION =====
cat("Initial data inspection...\n")
cat(sprintf("  Dimensions: %d rows x %d columns\n", nrow(df), ncol(df)))
cat(sprintf("  Variables: %s\n", paste(names(df), collapse=", ")))
cat("\n")

# ===== 3. DATA CLEANING =====

original_rows <- nrow(df)

# Convert to factors
df$Age <- factor(df$Age, levels = c("Puppy", "Fully grown"), ordered = TRUE)
df$Returned <- factor(df$Returned, levels = c("No", "Yes", "Unknown"))
df$Reason <- factor(df$Reason)

# Count missing values before removal
na_breeds <- sum(is.na(df$Breed))
invalid_rehomed <- sum(df$Rehomed == 99999 | df$Rehomed < 0, na.rm = TRUE)

# Remove missing values
df <- subset(df, !is.na(df$Breed))
df <- df[!(df$Rehomed == 99999 | df$Rehomed < 0), ]

# Print cleaning summary
cat("DATA CLEANING SUMMARY\n")
cat(sprintf("Original rows: %d\n", original_rows))
cat(sprintf("Removed - Missing Breed: %d\n", na_breeds))
cat(sprintf("Removed - Invalid Rehomed: %d\n", invalid_rehomed))
cat(sprintf("Total removed: %d (%.1f%%)\n",
            original_rows - nrow(df),
            ((original_rows - nrow(df))/original_rows)*100))
cat(sprintf("Final dataset: %d rows\n", nrow(df)))
cat(sprintf("\nBreeds in dataset: %d\n", length(unique(df$Breed))))
for (b in unique(df$Breed)) {
  cat(sprintf("  - %s: %d dogs (%.1f%%)\n",
              b, sum(df$Breed == b), (sum(df$Breed == b)/nrow(df))*100))
}

# ===== 4. DATA EXPLORATION =====

# Create breed subsets
breeds <- unique(df$Breed)
breed_subsets <- list()

for (breed in breeds) {
  name <- gsub(" +", "_", trimws(tolower(breed)))
  tmp <- subset(df, Breed == breed)
  rownames(tmp) <- NULL
  breed_subsets[[name]] <- tmp
}

# Overall summary
overall_summary <- data.frame(
```

```
  n = nrow(df),
  mean = round(mean(df$Rehomed), 2),
  median = round(median(df$Rehomed), 2),
  sd = round(sd(df$Rehomed), 2),
  min = min(df$Rehomed),
  max = max(df$Rehomed)
)

cat("OVERALL SUMMARY STATISTICS\n")
print(overall_summary)
cat("\n")

# Summary by breed
breed_summary <- aggregate(Rehomed ~ Breed, data = df,
                           FUN = function(x) c(
                             n = length(x),
                             mean = round(mean(x), 2),
                             median = round(median(x), 2),
                             sd = round(sd(x), 2),
                             min = min(x),
                             max = max(x)
                           ))

cat("SUMMARY BY BREED\n")
print(breed_summary)
cat("\n")

# Visualizations
cat("Creating exploratory plots...\n\n")

# Boxplot
boxplot(Rehomed ~ Breed, data = df,
        xlab = "Breed",
        ylab = "Rehoming Time (weeks)",
        main = "Rehoming Time by Breed",
        col = "lightblue")
abline(h = 27, col = "red", lty = 2, lwd = 2)
legend("topright", legend = "Previous research (27 weeks)",
       col = "red", lty = 2, lwd = 2)

# Histograms
par(mfrow = c(2, 2))
for (breed_name in names(breed_subsets)) {
  readable_name <- tools::toTitleCase(gsub("_", " ", breed_name))
  hist(breed_subsets[[breed_name]]$Rehomed,
       main = readable_name,
       xlab = "Rehoming Time (weeks)",
       col = "lightblue",
       border = "white",
       breaks = 15,
       xlim = c(0, 50))
  abline(v = 27, col = "red", lty = 2, lwd = 2)
}
par(mfrow = c(1, 1))

# Q-Q plots
par(mfrow = c(1, 3))
for (breed_name in names(breed_subsets)) {
  readable_name <- tools::toTitleCase(gsub("_", " ", breed_name))
  x <- breed_subsets[[breed_name]]$Rehomed
  qqnorm(x, main = paste("Normal Q-Q:", readable_name),
         ylab = "Sample quantiles")
  qqline(x, col = "red", lwd = 2)
}
par(mfrow = c(1, 1))

# ===== 5. DISTRIBUTION MODELING =====
```

```
fit_distribution <- function(data, breed_name) {
  paste(rep("=", 70), collapse="")
  cat("DISTRIBUTION MODELING:", breed_name, "\n")
  paste(rep("=", 70), collapse="")

  rehomed_times <- data$Rehomed
  n <- length(rehomed_times)
  mean_val <- mean(rehomed_times)
  sd_val <- sd(rehomed_times)

  cat(sprintf("Sample size: n = %d\n", n))
  cat(sprintf("Mean: %.2f weeks\n", mean_val))
  cat(sprintf("SD: %.2f weeks\n\n", sd_val))

  cat("Normal Distribution N(mu, sigma^2):\n")
  cat(sprintf("  mu-hat = %.2f weeks\n", mean_val))
  cat(sprintf("  sigma-hat = %.2f weeks\n\n", sd_val))

  # Shapiro-Wilk test
  if (n >= 3 && n <= 5000) {
    shapiro_test <- shapiro.test(rehomed_times)
    cat("Shapiro-Wilk Normality Test:\n")
    cat(sprintf("  W statistic = %.4f\n", shapiro_test$statistic))

    if (shapiro_test$p.value < 0.001) {
      cat(sprintf("  p-value < 0.001\n\n"))
    } else {
      cat(sprintf("  p-value = %.4f\n\n", shapiro_test$p.value))
    }

    cat("Interpretation:\n")
    if (shapiro_test$p.value > 0.05) {
      cat("  Data is consistent with normality (p > 0.05)\n")
    } else {
      cat("  Data departs from normality (p < 0.05)\n")
      if (n > 100) {
        cat("  Note: Large samples make test very sensitive.\n")
        cat("        Normal approximation may still be reasonable.\n")
      }
    }
  }

  cat("\n")

  return(list(
    n = n,
    mu = mean_val,
    sigma = sd_val,
    w_stat = shapiro_test$statistic,
    p_value = shapiro_test$p.value
  ))
}

cat("DISTRIBUTION MODELING ANALYSIS\n")

model_results <- list()
for (breed_name in names(breed_subsets)) {
  readable_name <- tools::toTitleCase(gsub("_", " ", breed_name))
  model_results[[breed_name]] <- fit_distribution(
    breed_subsets[[breed_name]],
    readable_name
  )
}

# Visualise fitted distributions
par(mfrow = c(2, 2))
for (breed_name in names(breed_subsets)) {
  readable_name <- tools::toTitleCase(gsub("_", " ", breed_name))
```

```
  data <- breed_subsets[[breed_name]]
  result <- model_results[[breed_name]]

  hist(data$Rehomed,
       probability = TRUE,
       breaks = 15,
       col = "lightblue",
       border = "white",
       main = readable_name,
       xlab = "Rehoming Time (weeks)",
       ylab = "Density")

  x <- seq(0, max(data$Rehomed), length.out = 200)
  lines(x, dnorm(x, mean = result$mu, sd = result$sigma),
        col = "red", lwd = 2)

  text(max(data$Rehomed) * 0.65,
       max(hist(data$Rehomed, plot=FALSE)$density) * 0.85,
       sprintf("N(%.1f, %.1f^2)\nW = %.3f\np = %.3f",
               result$mu, result$sigma, result$w_stat, result$p_value),
       cex = 0.75, pos = 4)

  legend("topright", legend = "Fitted Normal",
         col = "red", lwd = 2, cex = 0.8)
}
par(mfrow = c(1, 1))

# ===== 6. INFERENCE (CONFIDENCE INTERVALS) =====

calculate_ci <- function(data, conf_level = 0.95) {
  n <- nrow(data)
  mean_val <- mean(data$Rehomed)
  sd_val <- sd(data$Rehomed)
  se <- sd_val / sqrt(n)
  df <- n - 1

  alpha <- 1 - conf_level
  t_star <- qt(1 - alpha/2, df = df)

  ci_lower <- mean_val - t_star * se
  ci_upper <- mean_val + t_star * se
  margin_error <- t_star * se

  return(list(
    n = n,
    mean = mean_val,
    sd = sd_val,
    se = se,
    df = df,
    t_star = t_star,
    ci_lower = ci_lower,
    ci_upper = ci_upper,
    margin_error = margin_error,
    contains_27 = (ci_lower <= 27 & ci_upper >= 27)
  ))
}

ci_results <- lapply(breed_subsets, calculate_ci, conf_level = 0.95)

# Create summary table
ci_table <- data.frame(
  Breed = sapply(names(ci_results), function(x) tools::toTitleCase(gsub("_", " ", x))),
  n = sapply(ci_results, function(x) x$n),
  Mean = sapply(ci_results, function(x) round(x$mean, 2)),
  SD = sapply(ci_results, function(x) round(x$sd, 2)),
  SE = sapply(ci_results, function(x) round(x$se, 3)),
  df = sapply(ci_results, function(x) x$df),
  t_critical = sapply(ci_results, function(x) round(x$t_star, 3)),
```

```
  CI_Lower = sapply(ci_results, function(x) round(x$ci_lower, 2)),
  CI_Upper = sapply(ci_results, function(x) round(x$ci_upper, 2)),
  Contains_27 = sapply(ci_results, function(x) ifelse(x$contains_27, "Yes", "No"))
)

cat("CONFIDENCE INTERVALS FOR MEAN REHOMING TIME\n")
print(ci_table)

# Detailed results
cat("DETAILED RESULTS\n")

for (breed_name in names(ci_results)) {
  result <- ci_results[[breed_name]]
  readable_name <- tools::toTitleCase(gsub("_", " ", breed_name))

  cat(sprintf("%s:\n", readable_name))
  cat(sprintf("  n = %d, Mean = %.2f weeks, SD = %.2f weeks\n",
              result$n, result$mean, result$sd))
  cat(sprintf("  95%% CI: (%.2f, %.2f) weeks\n", result$ci_lower, result$ci_upper))

  if (!result$contains_27) {
    if (result$ci_upper < 27) {
      cat("  -> Significantly LOWER than 27 weeks (p < 0.05)\n")
    } else {
      cat("  -> Significantly HIGHER than 27 weeks (p < 0.05)\n")
    }
  } else {
    cat("  -> NOT significantly different from 27 weeks\n")
  }
  cat("\n")
}

# Visualise CIs
plot_ci <- function(ci_table) {
  par(mar = c(5, 10, 4, 2))
  ci_table <- ci_table[order(ci_table$Mean), ]

  plot(ci_table$Mean, 1:nrow(ci_table),
       xlim = c(min(ci_table$CI_Lower) - 2, 30),
       ylim = c(0.5, nrow(ci_table) + 0.5),
       xlab = "Rehoming Time (weeks)",
       ylab = "",
       yaxt = "n",
       main = "95% Confidence Intervals for Mean Rehoming Time",
       pch = 19, cex = 1.5, col = "darkblue")

  axis(2, at = 1:nrow(ci_table), labels = ci_table$Breed, las = 1)

  for (i in 1:nrow(ci_table)) {
    lines(c(ci_table$CI_Lower[i], ci_table$CI_Upper[i]), c(i, i),
          lwd = 3, col = "darkblue")
    points(c(ci_table$CI_Lower[i], ci_table$CI_Upper[i]), c(i, i),
           pch = "|", cex = 2, col = "darkblue")
  }

  abline(v = 27, col = "red", lwd = 2, lty = 2)
  text(27, nrow(ci_table) + 0.3, "Previous\nResearch\n(27 weeks)",
       col = "red", cex = 0.9)
  grid(nx = NULL, ny = NA, col = "lightgray", lty = "dotted")
}

plot_ci(ci_table)

# ===== 7. COMPARISON BETWEEN BREEDS =====

compare_breeds <- function(data1, data2, breed1_name, breed2_name) {
  paste(rep("=", 70), collapse="")
  cat(sprintf("%s vs %s\n", breed1_name, breed2_name))
```

```
  paste(rep("=", 70), collapse="")

  # Welch's t-test (unequal variances)
  test_result <- t.test(data1$Rehomed, data2$Rehomed,
                        var.equal = FALSE,
                        conf.level = 0.95)

  cat("Method: Welch's t-test (unequal variances assumed)\n")
  cat(sprintf("95%% CI for difference (mu1 - mu2): (%.2f, %.2f) weeks\n",
              test_result$conf.int[1], test_result$conf.int[2]))
  cat(sprintf("Mean difference: %.2f weeks\n",
              mean(data1$Rehomed) - mean(data2$Rehomed)))

  if (test_result$conf.int[1] <= 0 && test_result$conf.int[2] >= 0) {
    cat("Conclusion: No significant difference (CI contains 0)\n")
  } else {
    cat("Conclusion: Significant difference (CI excludes 0)\n")
  }

  return(test_result)
}

cat("PAIRWISE BREED COMPARISONS\n")

breed_names_list <- names(breed_subsets)
comparison_results <- list()

for (i in 1:(length(breed_names_list)-1)) {
  for (j in (i+1):length(breed_names_list)) {
    breed1 <- breed_names_list[i]
    breed2 <- breed_names_list[j]

    name1 <- tools::toTitleCase(gsub("_", " ", breed1))
    name2 <- tools::toTitleCase(gsub("_", " ", breed2))

    comparison_name <- paste(breed1, "vs", breed2, sep = "_")

    comparison_results[[comparison_name]] <- compare_breeds(
      breed_subsets[[breed1]],
      breed_subsets[[breed2]],
      name1,
      name2
    )
  }
}

# Comparison summary table
comparison_table <- data.frame(
  Comparison = character(),
  Mean_Diff = numeric(),
  CI_Lower = numeric(),
  CI_Upper = numeric(),
  Significant = character(),
  stringsAsFactors = FALSE
)

for (comp_name in names(comparison_results)) {
  result <- comparison_results[[comp_name]]

  readable_comp <- gsub("_", " ", comp_name)
  readable_comp <- gsub(" vs ", " vs ", tools::toTitleCase(readable_comp))

  is_sig <- !(result$conf.int[1] <= 0 && result$conf.int[2] >= 0)

  comparison_table <- rbind(comparison_table, data.frame(
    Comparison = readable_comp,
    Mean_Diff = round(diff(result$estimate), 2),
    CI_Lower = round(result$conf.int[1], 2),
```

```
    CI_Upper = round(result$conf.int[2], 2),
    Significant = ifelse(is_sig, "Yes", "No")
  ))
}

cat("COMPARISON SUMMARY\n")
print(comparison_table)

# ===== 8. EXPORT RESULTS =====

write.csv(ci_table, "confidence_intervals.csv", row.names = FALSE)
write.csv(comparison_table, "breed_comparisons.csv", row.names = FALSE)

cat("ANALYSIS COMPLETE\n")
cat("Results exported to:\n")
cat("  - confidence_intervals.csv\n")
cat("  - breed_comparisons.csv\n")
```