MATH185, HW2

Victor Pekkari — epekkari@ucsd.edu

February 5, 2025

Problem 1

```
> source("/Users/victorpekkari/Desktop/comp_stats/hw2/q1.r", encoding = "UTF-8")
Question 1a)   Estimated θ̂: 0.9842783
Question 1b)   Estimated bias of the estimator: -0.3370668
Question 1c)   Estimated variance of the estimator: 2.560838
Question 1d)   95% Percentile Bootstrap CI: [ -3.076798 , 3.101148 ]
Question 1e)   95% Hall's Percentile Bootstrap CI: [ -1.132591 , 5.045354 ]
```

Figure 1: output for question 1

Code for problem 1:

```r
setwd("/Users/victorpekkari/Desktop/comp_stats/hw2")
data <- read.csv("lightbulb.csv")

brand_a <- data$times[data$brand == "A"]
brand_b <- data$times[data$brand == "B"]

theta_hat <- median(brand_a) - median(brand_b)
cat("Question 1a)   Estimated     :", theta_hat, "\n")

# Number of bootstrap samples
b_bias <- 200
b_var <- 200
b_ci <- 1000

bootstrapped_median <- function(data_a, data_b, iterations) {
  median_diff <- numeric(iterations)
  for (i in 1:iterations) {
    resample_a <- sample(data_a, replace = TRUE)
    resample_b <- sample(data_b, replace = TRUE)
    median_diff[i] <- median(resample_a) - median(resample_b)
  }
```

```
  return(median_diff)
}

bootstrap_diffs_bias <- bootstrapped_median(brand_a, brand_b, b_bias)
bias <- mean(bootstrap_diffs_bias) - theta_hat
cat("Question 1b) - Estimated bias of the estimator:", bias, "\n")

bootstrap_diffs_var <- bootstrapped_median(brand_a, brand_b, b_var)
variance <- var(bootstrap_diffs_var)
cat("Question 1c) - Estimated variance of the estimator:", variance, "\n")

bootstrap_diffs_ci <- bootstrapped_median(brand_a, brand_b, b_ci)
quantiles <- quantile(bootstrap_diffs_ci, probs = c(0.025, 0.975))
cat("Question 1d) - 95% Percentile Bootstrap CI: [", quantiles[1], ",", quantile

ci_hall <- 2 * theta_hat - quantile(bootstrap_diffs_ci, probs = c(0.975, 0.025))
cat("Question 1e) - 95% Halls Percentile Bootstrap CI: [", ci_hall[1], ",", c
```

Problem 2

```
> source("/Users/victorpekkari/Desktop/comp_stats/hw2/q2.r", encoding = "UTF-8")
Estimated Type I error (n = m = 10): 0.054
Estimated Type I error (n = m = 50): 0.043
Estimated Power (n = m = 10): 0.267
Estimated Power (n = m = 50): 0.809
```

Figure 2: output for question 2

(a)

Comment: Choosing $\alpha = 0.05$ should mean that we have a type I error rate of 5% which we do, since we reject the null hypothesis about 5% of the time.

(b)

Comment: My type I error rate barely changed, and theoretically it shoudn't change.

(d)

Comment: The power increases significantly when the number of samples is increased. This is theoretically correct as the type II error (β) should decrease and therefore the power ($1 - \beta$) should increase. I think it is because as the sample size increases the standard error decreases.

Code for problem 2

```r
#set.seed(123)  # Ensure reproducibility

perm_test <- function(x, y, b = 1000) {
  t_obs <- mean(x) - mean(y)  # observed test statistic
  combined <- c(x, y)
  n <- length(x)

  t_perm <- replicate(b, {
    perm <- sample(combined)  # Shuffle the combined data
    mean(perm[1:n]) - mean(perm[(n + 1):length(combined)])
  })

  p_value <- (sum(t_perm >= t_obs) + 1) / (b + 1)
  return(p_value)
}

estimate_type1_error <- function(n = 10, m = 10,
                                 b = 1000, alpha = 0.05, sims = 1000) {
  type1_errors <- replicate(sims, {
    x <- rnorm(n)
    y <- rnorm(m)
    p_val <- perm_test(x, y, b)
    return(p_val < alpha)
  })
  return(mean(type1_errors))  # Proportion of false rejections
}

estimate_power <- function(n = 10, m = 10,
                           b = 1000, alpha = 0.05, sims = 1000) {
  power <- replicate(sims, {
    x <- rnorm(n, mean = 0.5, sd = 1)  # Shifted mean for X
    y <- rnorm(m, mean = 0, sd = 1)    # Y remains standard normal
    p_val <- perm_test(x, y, b)
    return(p_val < alpha)
  })
  return(mean(power))  # Proportion of true rejections
}

type1_error_10 <- estimate_type1_error(n = 10, m = 10, b = 1000, sims = 1000)
cat("Estimated Type I error (n = m = 10):", type1_error_10, "\n")

type1_error_50 <- estimate_type1_error(n = 50, m = 50, b = 1000, sims = 1000)
cat("Estimated Type I error (n = m = 50):", type1_error_50, "\n")
```

```
power_10 <- estimate_power(n = 10, m = 10, b = 1000, sims = 1000)
cat("Estimated Power (n = m = 10):", power_10, "\n")

power_50 <- estimate_power(n = 50, m = 50, b = 1000, sims = 1000)
cat("Estimated Power (n = m = 50):", power_50, "\n")
```

Problem 3

```
> source("/Users/victorpekkari/Desktop/comp_stats/hw2/q3.r",
Estimated Type I Error: 0.094
Estimated Power for a=0.8: 0.738
Estimated Power for a=1: 0.871
Estimated Power for a=1.2: 0.963
```

Figure 3: output for question 3

(a)

Comment: Yes it is very close to the theoretical 10%.

(b)

$$Y = aX + \epsilon$$

Comment: The power increases as "a" increases which is expected, since larger "a" means stronger linear correlation. The larger "a" makes it easier to distinguish the true relationship (a) from noise (ϵ).

Code for problem 3

```
set.seed(123)

perm_test <- function(x, y, b) {
  correlation <- cor(x, y)
  permuted_corrs <- numeric(b)

  for (i in 1:b) {
    y_perm <- sample(y)
    permuted_corrs[i] <- cor(x, y_perm)
  }

  p_value <- mean(abs(permuted_corrs) >= abs(correlation))
```

```
  return(p_value)
}

estimate_type1_error <- function(sims, alpha) {
  type1_errors <- replicate(sims, {
    x <- rnorm(100)
    y <- rnorm(100)
    p_val <- perm_test(x, y, b = 1000)
    return(p_val < alpha)
  })

  return(mean(type1_errors))
}

estimate_power <- function(sims, alpha, a_values) {
  power_estimates <- sapply(a_values, function(a) {
    rejections <- replicate(sims, {
      x <- runif(100, 0, 1)
      epsilon <- rnorm(100)
      y <- a * x + epsilon
      p_val <- perm_test(x, y, b = 1000)
      return(p_val < alpha)
    })

    return(mean(rejections))
  })

  return(setNames(power_estimates, a_values))
}

# Run simulations
type_I_error <- estimate_type1_error(1000, 0.10)
a_values <- c(0.8, 1.0, 1.2)
power_results <- estimate_power(1000, 0.10, a_values)

cat(sprintf("Estimated Type I Error: %.3f\n", type_I_error))
for (a in names(power_results)) {
  cat(sprintf("Estimated Power for a=%s: %.3f\n", a, power_results[[a]]))
}
```