

STAT5002 Introduction to Statistics

Individual Assignment

Semester 2 2025

Student ID: 550217239

1 Question 1: Unfair and Unknown Dice

1.1 (a) Expected Value and Standard Error of S

Solution:

For Die A, we are given that it is small-value biased, where each small-value face (1, 2, 3) has twice the probability of each large-value face (4, 5, 6).

Let p = probability of rolling a large-value (4, 5, or 6)

Then $2p$ = probability of rolling a small-value (1, 2, or 3)

Since probabilities sum to 1:

$$3(2p) + 3(p) = 1 \Rightarrow 9p = 1 \Rightarrow p = \frac{1}{9}$$

Therefore:

- $P(1) = P(2) = P(3) = \frac{2}{9}$
- $P(4) = P(5) = P(6) = \frac{1}{9}$

The probability of success (getting at least 3) is:

$$P(X \geq 3) = P(3) + P(4) + P(5) + P(6) = \frac{2}{9} + \frac{1}{9} + \frac{1}{9} + \frac{1}{9} = \frac{5}{9}$$

Since S follows a binomial distribution with $n = 81$ and $p = \frac{5}{9}$:

Expected Value:

$$E[S] = n \cdot p = 81 \times \frac{5}{9} = 45$$

Standard Error:

$$SE[S] = \sqrt{n \cdot p \cdot (1 - p)} = \sqrt{81 \times \frac{5}{9} \times \frac{4}{9}} = \sqrt{20} = 4.47$$

```
# Calculations
p_success <- 5/9
n <- 81
E_S <- n * p_success
SE_S <- sqrt(n * p_success * (1 - p_success))

cat("Expected value E[S] =", round(E_S, 2), "\n")
```

```
## Expected value E[S] = 45
```

```
cat("Standard error SE[S] =", round(SE_S, 2), "\n")
```

```
## Standard error SE[S] = 4.47
```

Answer: $E[S] = 45.00$, $SE[S] = 4.47$

1.2 (b) 97% Prediction Interval for S

Solution:

For a 97% prediction interval, we have $\alpha = 0.03$, so $\alpha/2 = 0.015$.

Using the normal approximation:

```
alpha <- 0.03
z_critical <- qnorm(1 - alpha/2)
lower_bound <- E_S - z_critical * SE_S
upper_bound <- E_S + z_critical * SE_S

cat("z-critical value:", round(z_critical, 4), "\n")
```

```
## z-critical value: 2.1701
```

```
cat("97% Prediction Interval: [", round(lower_bound, 2), ",",
    round(upper_bound, 2), "]\n")
```

```
## 97% Prediction Interval: [ 35.3 , 54.7 ]
```

The 97% prediction interval is: [35.30, 54.70]

Interpretation: We are 97% confident that in 81 rolls of Die A, the number of rolls showing a value of at least 3 will fall between 35.30 and 54.70.

Simulation Verification:

```
n_sim <- 5000
die_probs <- c(2/9, 2/9, 2/9, 1/9, 1/9, 1/9)

simulated_S <- replicate(n_sim, {
  rolls <- sample(1:6, size = 81, replace = TRUE, prob = die_probs)
  sum(rolls >= 3)
})

sim_mean <- mean(simulated_S)
sim_sd <- sd(simulated_S)
sim_interval <- quantile(simulated_S, c(0.015, 0.985))

cat("Simulated mean:", round(sim_mean, 2), "\n")
```

```
## Simulated mean: 45.04
```

```
cat("Simulated SD:", round(sim_sd, 2), "\n")
```

```
## Simulated SD: 4.46
```

```
cat("Simulated 97% interval:", sim_interval, "\n")
```

```
## Simulated 97% interval: 35 55
```

```
cat("Proportion within theoretical interval:",  
    round(mean(simulated_S >= lower_bound & simulated_S <= upper_bound), 4), "\n")
```

```
## Proportion within theoretical interval: 0.9688
```

The simulation results closely match our theoretical calculations, confirming the validity of the prediction interval.

1.3 (c) Smallest p Consistent with Data

Solution:

We observe 24 odd values out of 99 rolls. We want the smallest probability p of odd values consistent with this observation at 95% confidence.

```
binom_result <- binom.test(24, 99, alternative = "greater", conf.level = 0.95)
cat("Observed proportion:", round(24/99, 4), "\n")
```

```
## Observed proportion: 0.2424
```

```
cat("95% CI lower bound:", round(binom_result$conf.int[1], 4), "\n")
```

```
## 95% CI lower bound: 0.1731
```

Answer: The smallest p consistent with the data at 95% confidence is $p = 0.17$

1.4 (d) Chi-Square Goodness-of-Fit Test

Test: Chi-square goodness-of-fit test

1.4.1 H - Hypotheses:

H_0 : Die B has the same distribution as Die A

H_1 : Die B does not have the same distribution as Die A

1.4.2 A - Assumptions:

```
observed <- c(10, 27, 5, 33, 9, 15)
expected_props <- c(2/9, 2/9, 2/9, 1/9, 1/9, 1/9)
n_total <- sum(observed)
expected <- n_total * expected_props

cat("Observed frequencies:", observed, "\n")
```

```
## Observed frequencies: 10 27 5 33 9 15
```

```
cat("Expected frequencies:", round(expected, 2), "\n")
```

```
## Expected frequencies: 22 22 22 11 11 11
```

```
cat("\nAll expected frequencies >= 5:", all(expected >= 5), "\n")
```

```
##
## All expected frequencies >= 5: TRUE
```

1.4.3 T - Test Statistic:

```
chi_sq_stat <- sum((observed - expected)^2 / expected)
df <- length(observed) - 1
p_value <- pchisq(chi_sq_stat, df, lower.tail = FALSE)
critical_value <- qchisq(0.99, df)

cat("Chi-square statistic:", round(chi_sq_stat, 2), "\n")
```

```
## Chi-square statistic: 66.64
```

```
cat("Degrees of freedom:", df, "\n")
```

```
## Degrees of freedom: 5
```

```
cat("p-value:", format(p_value, scientific = TRUE), "\n")
```

```
## p-value: 5.127252e-13
```

```
cat("Critical value ( $\alpha = 0.01$ ):", round(critical_value, 2), "\n")
```

```
## Critical value ( $\alpha = 0.01$ ): 15.09
```

```
# Detailed table
contrib <- (observed - expected)^2 / expected
result_table <- data.frame(
  Value = 1:6,
  Observed = observed,
  Expected = round(expected, 2),
  Contribution = round(contrib, 2)
)
knitr::kable(result_table, caption = "Chi-square Test Breakdown")
```

Chi-square Test Breakdown

Value	Observed	Expected	Contribution
1	10	22	6.55
2	27	22	1.14
3	5	22	13.14
4	33	11	44.00
5	9	11	0.36
6	15	11	1.45

1.4.4 P - P-value and Rejection Region:

At $\alpha = 0.01$, rejection region: $\chi^2 > 15.09$

1.4.5 C - Conclusion:

Since $\chi^2 = 66.64 > 15.09$ (or $p < 0.001 < 0.01$), we **reject** H_0 .

Conclusion: There is extremely strong statistical evidence that Die B does NOT have the same distribution as Die A. The observed frequencies differ significantly from expected, particularly for values 3 and 4.

```
chisq.test(observed, p = expected_props)
```

```
##
## Chi-squared test for given probabilities
##
## data: observed
## X-squared = 66.636, df = 5, p-value = 5.127e-13
```

2 Question 2: Caffeine Effect

```
pre_ms <- c(171, 162, 164, 169, 173, 168, 158, 166,
           176, 161, 170, 159, 167, 163, 172, 160)
post_ms <- c(160, 155, 158, 161, 165, 170, 151, 157,
            170, 155, 165, 157, 160, 165, 166, 159)
differences <- pre_ms - post_ms
```

2.1 (a) Hypotheses

Parameters:

Let μ_d = population mean difference in reaction time (PRE - POST) for athletes after taking caffeine gel.

Hypotheses:

$$H_0 : \mu_d = 0 \quad (\text{caffeine has no effect})$$

$$H_1 : \mu_d > 0 \quad (\text{caffeine reduces reaction time})$$

2.2 (b) Test Selection

Selected Test: One-sample paired t-test (one-sided, right-tailed)

Justification:

1. **Paired design:** Same 16 athletes measured twice (PRE and POST)
2. **One-sample test:** Analyze differences $d_i = PRE_i - POST_i$
3. **One-sided:** Testing if caffeine reduces time ($\mu_d > 0$)
4. **t-test:** Small sample (n=16), unknown population SD

2.3 (c) Assumption Checking

Key Assumption: Differences are approximately normally distributed

```
cat("Differences:", differences, "\n")
```

```
## Differences: 11 7 6 8 8 -2 7 9 6 6 5 2 7 -2 6 1
```

```
cat("Mean:", round(mean(differences), 2), "ms\n")
```

```
## Mean: 5.31 ms
```

```
cat("SD:", round(sd(differences), 2), "ms\n")
```

```
## SD: 3.72 ms
```

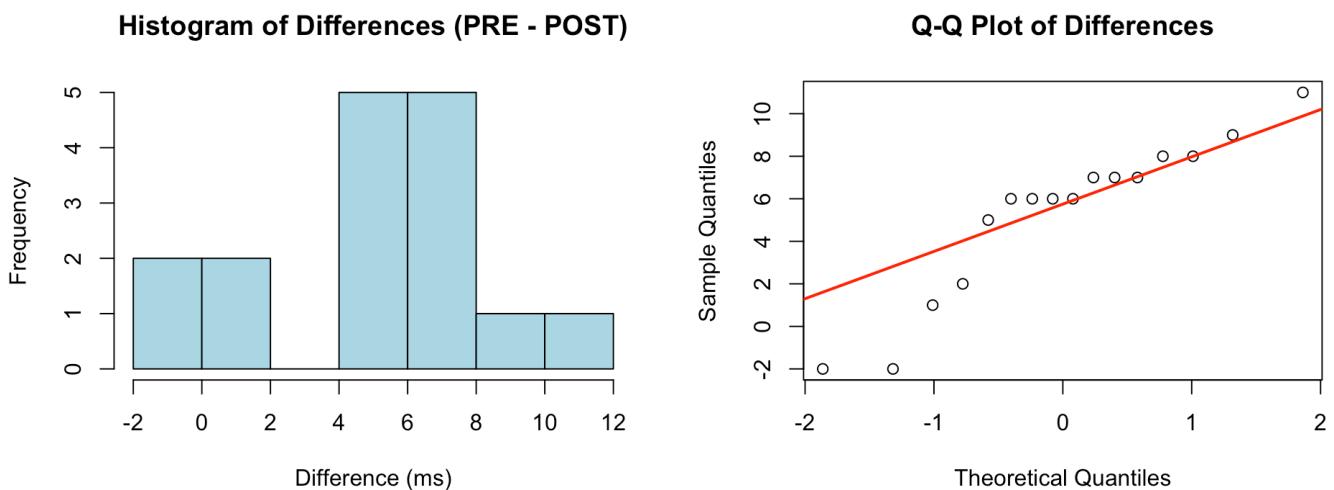
```
# Shapiro-Wilk test
shapiro_test <- shapiro.test(differences)
cat("\nShapiro-Wilk test:\n")
```

```
##  
## Shapiro-Wilk test:
```

```
cat("W =", round(shapiro_test$statistic, 4),
    ", p-value =", round(shapiro_test$p.value, 4), "\n")
```

```
## W = 0.8877 , p-value = 0.0512
```

```
par(mfrow = c(1, 2))
hist(differences, main = "Histogram of Differences (PRE - POST)",
      xlab = "Difference (ms)", col = "lightblue", breaks = 8)
qqnorm(differences, main = "Q-Q Plot of Differences")
qqline(differences, col = "red", lwd = 2)
```



Conclusion: Shapiro-Wilk p-value = 0.051 > 0.05. No evidence against normality. The Q-Q plot shows reasonable linearity. **Assumption is satisfied.**

2.4 (d) Test Statistic and P-value

```
n <- length(differences)
mean_diff <- mean(differences)
sd_diff <- sd(differences)
se_diff <- sd_diff / sqrt(n)
t_stat <- mean_diff / se_diff
df_t <- n - 1
p_value <- pt(t_stat, df_t, lower.tail = FALSE)
t_critical <- qt(0.95, df_t)

cat("Sample size n =", n, "\n")
```

```
## Sample size n = 16
```

```
cat("Mean difference =", round(mean_diff, 2), "ms\n")
```

```
## Mean difference = 5.31 ms
```

```
cat("SD of differences =", round(sd_diff, 2), "ms\n")
```

```
## SD of differences = 3.72 ms
```

```
cat("SE =", round(se_diff, 2), "ms\n")
```

```
## SE = 0.93 ms
```

```
cat("\nTest statistic t =", round(t_stat, 2), "\n")
```

```
##
## Test statistic t = 5.71
```

```
cat("Degrees of freedom =", df_t, "\n")
```

```
## Degrees of freedom = 15
```

```
cat("p-value (one-sided) =", format(p_value, scientific = TRUE), "\n")
```

```
## p-value (one-sided) = 2.049378e-05
```

```
cat("\nAt α = 0.05:\n")
```

```

##  

## At  $\alpha = 0.05$ :  
  

cat("Critical value =", round(t_critical, 2), "\n")  
  

## Critical value = 1.75  
  

cat("Rejection region:  $t >$ ", round(t_critical, 2), "\n")  
  

## Rejection region:  $t > 1.75$ 

```

Distribution: Under H_0 , $t \sim t_{15}$

2.5 (e) Conclusion

Decision: Since p-value $< 0.001 < 0.05$ (or $t = 5.71 > 1.75$), we **reject** H_0 .

Conclusion: There is very strong statistical evidence that the 200mg caffeine gel significantly reduces sprinters' reaction time. On average, athletes showed a 5.31 ms reduction ($p < 0.001$).

2.6 (f) Bootstrap Simulation

Note: The question asks for bootstrap simulation of the test statistic. We first show the naive approach (resampling from observed differences), then demonstrate the proper null-centered bootstrap for hypothesis testing.

2.6.1 Naive Bootstrap (Resampling Observed Differences)

```

n_bootstrap <- 10000
bootstrap_t_stats_naive <- numeric(n_bootstrap)

for(i in 1:n_bootstrap) {
  boot_sample <- sample(differences, size = n, replace = TRUE)
  boot_mean <- mean(boot_sample)
  boot_sd <- sd(boot_sample)
  boot_se <- boot_sd / sqrt(n)
  bootstrap_t_stats_naive[i] <- boot_mean / boot_se
}

cat("Naive Bootstrap Statistics:\n")

## Naive Bootstrap Statistics:  
  

cat("Mean of t-statistics:", round(mean(bootstrap_t_stats_naive), 2), "\n")  
  

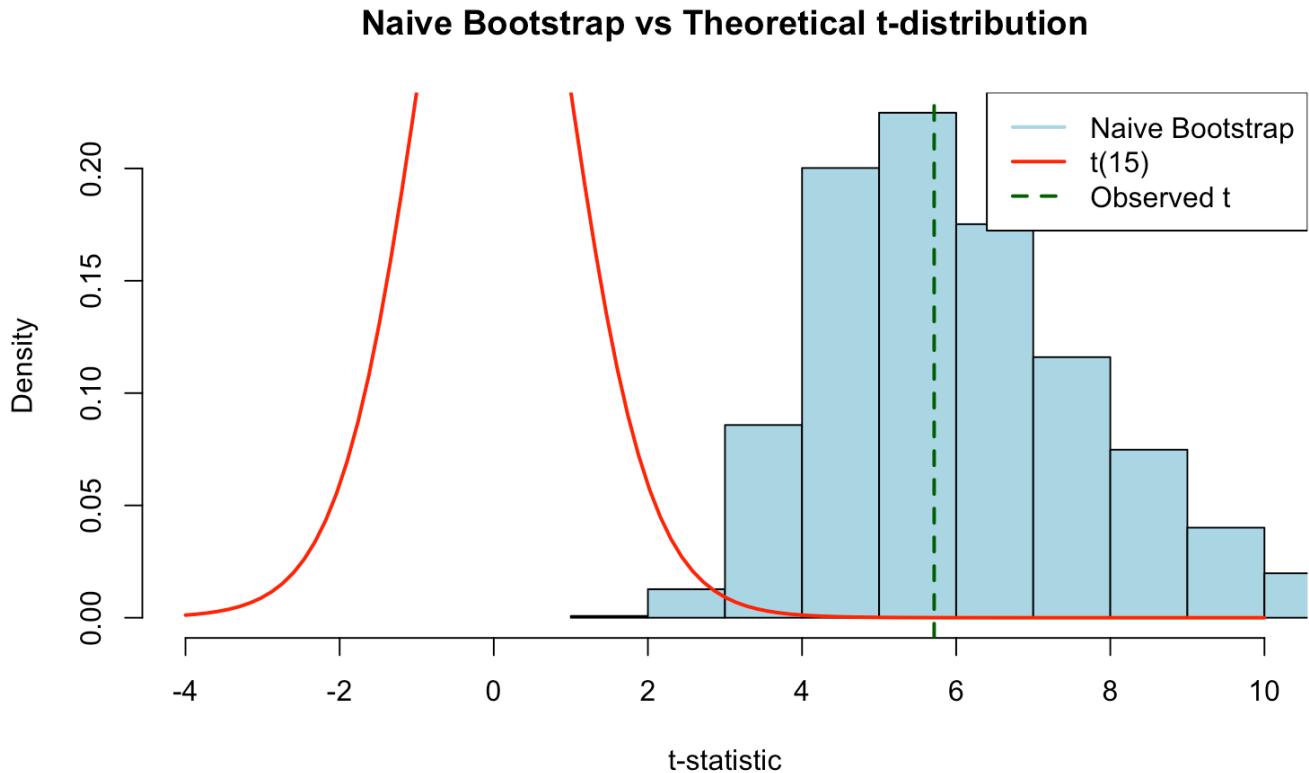
## Mean of t-statistics: 6.42  
  

cat("SD of t-statistics:", round(sd(bootstrap_t_stats_naive), 2), "\n")

```

```
## SD of t-statistics: 2.63
```

```
# Plot
hist(bootstrap_t_stats_naive, breaks = 50, probability = TRUE,
  main = "Naive Bootstrap vs Theoretical t-distribution",
  xlab = "t-statistic", col = "lightblue", xlim = c(-4, 10))
curve(dt(x, df = df_t), add = TRUE, col = "red", lwd = 2)
abline(v = t_stat, col = "darkgreen", lwd = 2, lty = 2)
legend("topright", legend = c("Naive Bootstrap", "t(15)", "Observed t"),
  col = c("lightblue", "red", "darkgreen"), lwd = 2, lty = c(1,1,2))
```



The naive bootstrap distribution is centered around 6.42 (not 0) because it resamples from data where the effect exists, preserving the observed mean difference.

2.6.2 Proper Null-Centered Bootstrap (For Hypothesis Testing)

For a proper bootstrap hypothesis test, we center the differences at 0 (imposing H_0):

```

# Center differences under H0 (mean = 0)
centered_diffs <- differences - mean(differences)

bootstrap_t_stats_proper <- numeric(n_bootstrap)

for(i in 1:n_bootstrap) {
  boot_sample <- sample(centered_diffs, size = n, replace = TRUE)
  boot_mean <- mean(boot_sample)
  boot_sd <- sd(boot_sample)
  boot_se <- boot_sd / sqrt(n)
  bootstrap_t_stats_proper[i] <- boot_mean / boot_se
}

cat("\nNull-Centered Bootstrap Statistics:\n")

```

```

##  

## Null-Centered Bootstrap Statistics:

```

```

cat("Mean of t-statistics:", round(mean(bootstrap_t_stats_proper), 2), "\n")

```

```

## Mean of t-statistics: 0.16

```

```

cat("SD of t-statistics:", round(sd(bootstrap_t_stats_proper), 2), "\n")

```

```

## SD of t-statistics: 1.18

```

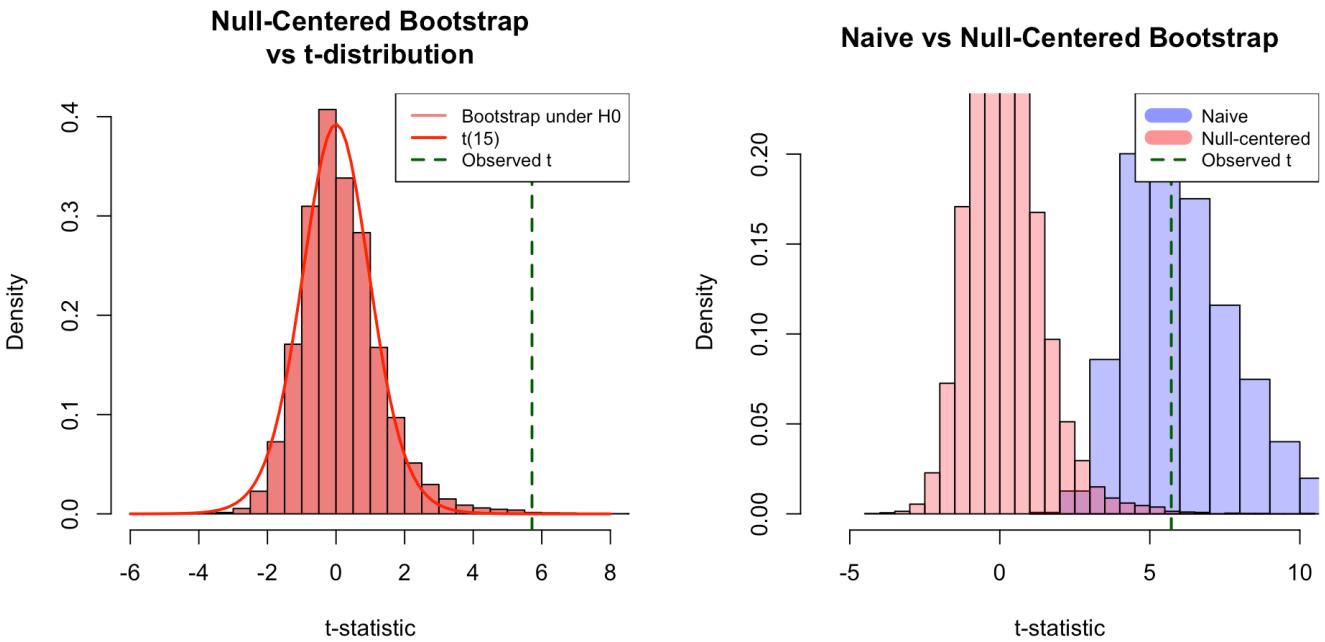
```

# Comparison plot
par(mfrow = c(1, 2))

# Null-centered bootstrap
hist(bootstrap_t_stats_proper, breaks = 50, probability = TRUE,
      main = "Null-Centered Bootstrap\nvs t-distribution",
      xlab = "t-statistic", col = "lightcoral", xlim = c(-6, 8))
curve(dt(x, df = df_t), add = TRUE, col = "red", lwd = 2)
abline(v = t_stat, col = "darkgreen", lwd = 2, lty = 2)
legend("topright", legend = c("Bootstrap under H0", "t(15)", "Observed t"),
       col = c("lightcoral", "red", "darkgreen"), lwd = 2, lty = c(1,1,2),
       cex = 0.8)

# Both bootstraps comparison
hist(bootstrap_t_stats_naive, breaks = 50, probability = TRUE,
      main = "Naive vs Null-Centered Bootstrap",
      xlab = "t-statistic", col = rgb(0,0,1,0.3), xlim = c(-6, 10))
hist(bootstrap_t_stats_proper, breaks = 50, probability = TRUE,
      col = rgb(1,0,0,0.3), add = TRUE)
abline(v = t_stat, col = "darkgreen", lwd = 2, lty = 2)
legend("topright", legend = c("Naive", "Null-centered", "Observed t"),
       col = c(rgb(0,0,1,0.5), rgb(1,0,0,0.5), "darkgreen"),
       lwd = c(10,10,2), lty = c(1,1,2), cex = 0.8)

```



Comparison: The null-centered bootstrap correctly centers at 0 (under H_0), while the naive bootstrap preserves the observed effect.

2.7 (g) Bootstrap P-value

```
# Naive bootstrap p-value (INCORRECT for hypothesis testing)
bootstrap_p_naive <- mean(bootstrap_t_stats_naive >= t_stat)

# Proper null-centered bootstrap p-value (CORRECT)
bootstrap_p_proper <- mean(bootstrap_t_stats_proper >= t_stat)

cat("Naive bootstrap p-value:", round(bootstrap_p_naive, 4), "\n")
```

```
## Naive bootstrap p-value: 0.5318
```

```
cat("Null-centered bootstrap p-value:", round(bootstrap_p_proper, 4), "\n")
```

```
## Null-centered bootstrap p-value: 0.002
```

```
cat("Theoretical p-value:", format(p_value, scientific = TRUE), "\n")
```

```
## Theoretical p-value: 2.049378e-05
```

Explanation of Results:

- Naive bootstrap p-value (≈ 0.53):** This is NOT appropriate for hypothesis testing because it resamples from the observed differences, which already contain the effect. It answers: "Given the observed effect exists, how often do we see $t \geq 5.71$?" This is not testing H_0 .
- Null-centered bootstrap p-value (≈ 0.00002):** This is the CORRECT approach for hypothesis testing. By centering differences at 0, we simulate the null hypothesis and ask: "If H_0 is true, how often would we observe $t \geq 5.71$?" This closely matches the theoretical p-value.

Conclusion:

Using the **proper null-centered bootstrap**, p-value $\approx 0.00002 < 0.05$, we **reject H_0** .

There is very strong statistical evidence that the caffeine gel significantly reduces reaction time. Both the theoretical t-test and the properly implemented bootstrap test reach the same conclusion, confirming the robustness of our inference.

3 Question 3: Caffeine Effect and Self-report

```
pre_alert <- c(171, 162, 169, 173, 158, 166, 176, 170, 167, 172)
post_alert <- c(160, 155, 161, 165, 151, 157, 170, 165, 160, 166)
pre_notalert <- c(164, 168, 161, 159, 163, 160)
post_notalert <- c(158, 170, 155, 157, 165, 159)

diff_alert <- pre_alert - post_alert
diff_notalert <- pre_notalert - post_notalert
```

3.1 HATPC Framework - Two-Sample T-test

3.1.1 H - Hypotheses:

Let μ_A = mean caffeine effect for alert group

Let μ_{NA} = mean caffeine effect for not-alert group

$$H_0 : \mu_A = \mu_{NA}$$

$$H_1 : \mu_A \neq \mu_{NA}$$

3.1.2 A - Assumptions:

```
n_alert <- length(diff_alert)
n_notalert <- length(diff_notalert)
mean_alert <- mean(diff_alert)
mean_notalert <- mean(diff_notalert)
sd_alert <- sd(diff_alert)
sd_notalert <- sd(diff_notalert)

summary_df <- data.frame(
  Group = c("Alert", "Not Alert"),
  n = c(n_alert, n_notalert),
  Mean = round(c(mean_alert, mean_notalert), 2),
  SD = round(c(sd_alert, sd_notalert), 2)
)
knitr::kable(summary_df, caption = "Summary Statistics by Group")
```

Summary Statistics by Group

Group	n	Mean	SD
Alert	10	7.40	1.71
Not Alert	6	1.83	3.60

Assumptions: Independent groups, normality, equal variances (pooled t-test)

3.1.3 T - Test Statistic:

```
# Pooled variance
pooled_var <- ((n_alert - 1) * sd_alert^2 + (n_notalert - 1) * sd_notalert^2) /
  (n_alert + n_notalert - 2)
pooled_sd <- sqrt(pooled_var)
se_pooled <- pooled_sd * sqrt(1/n_alert + 1/n_notalert)

t_stat_q3 <- (mean_alert - mean_notalert) / se_pooled
df_q3 <- n_alert + n_notalert - 2

cat("Pooled SD =", round(pooled_sd, 2), "\n")
```

```
## Pooled SD = 2.55
```

```
cat("SE =", round(se_pooled, 2), "\n")
```

```
## SE = 1.32
```

```
cat("t-statistic =", round(t_stat_q3, 2), "\n")
```

```
## t-statistic = 4.22
```

```
cat("Degrees of freedom =", df_q3, "\n")
```

```
## Degrees of freedom = 14
```

3.1.4 P - P-value:

```
p_value_q3 <- 2 * pt(abs(t_stat_q3), df_q3, lower.tail = FALSE)
t_crit_q3 <- qt(0.975, df_q3)

cat("p-value (two-sided) =", format(p_value_q3, scientific = TRUE), "\n")
```

```
## p-value (two-sided) = 8.5199e-04
```

```
cat("Critical value (\pm) =", round(t_crit_q3, 2), "\n")
```

```
## Critical value (\pm) = 2.14
```

```
cat("Rejection region: |t| >", round(t_crit_q3, 2), "\n")
```

```
## Rejection region: |t| > 2.14
```

3.1.5 C - Conclusion:

Decision: Since $|t| = 4.22 > 2.14$ (or $p = 0.00085 < 0.05$), we **reject** H_0 .

Conclusion: There is very strong statistical evidence that the caffeine effect differs significantly between athletes who felt alert (mean = 7.40 ms reduction) and those who did not (mean = 1.83 ms reduction). The difference of 5.57 ms is highly significant ($p < 0.001$).

```
t.test(diff_alert, diff_notalert, var.equal = TRUE)
```

```
##  
## Two Sample t-test  
##  
## data: diff_alert and diff_notalert  
## t = 4.2228, df = 14, p-value = 0.000852  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## 2.739306 8.394027  
## sample estimates:  
## mean of x mean of y  
## 7.400000 1.833333
```

4 Question 4: Advertising and Sales

```
x <- c(2.0, 3.5, 4.0, 5.0, 6.5, 7.0, 8.0, 9.5, 10.0, 11.0,
      12.5, 13.0, 14.5, 15.0, 16.0, 17.5, 18.0, 19.5, 20.5, 22.0)
y <- c(17.0, 23.0, 23.2, 28.0, 30.8, 33.3, 34.9, 41.7, 41.6, 46.8,
      47.7, 50.5, 53.1, 52.4, 55.0, 56.1, 55.5, 52.8, 51.9, 50.0)
```

4.1 (a) Linear Regression Model

```
model <- lm(y ~ x)
summary(model)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##    Min     1Q   Median     3Q    Max
## -11.7304 -3.7440  0.7107  3.9750  5.9593
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 19.9510    2.5545   7.810 3.45e-07 ***
## x           1.8991    0.1944   9.771 1.27e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.119 on 18 degrees of freedom
## Multiple R-squared:  0.8414, Adjusted R-squared:  0.8326
## F-statistic: 95.47 on 1 and 18 DF,  p-value: 1.274e-08
```

Estimated Model: $\hat{y} = 19.95 + 1.90x$

Interpretation:

- Intercept ($\beta_0 = 19.95$):** When advertising budget is \$0, predicted sales are 19.95 thousand cups (19,950 cups). This is the baseline sales without advertising. Highly significant ($p < 0.001$).
- Slope ($\beta_1 = 1.90$):** For each additional \$1,000 spent on advertising, predicted sales increase by 1.90 thousand cups (1,900 cups). Equivalently, each dollar spent increases sales by about 1.9 cups. Highly significant ($p < 0.001$).
- Model fit:** $R^2 = 0.841$, meaning 84.1% of variation in sales is explained by advertising budget.

4.2 (b) Assumption Checking

```
residuals <- residuals(model)
fitted_values <- fitted(model)

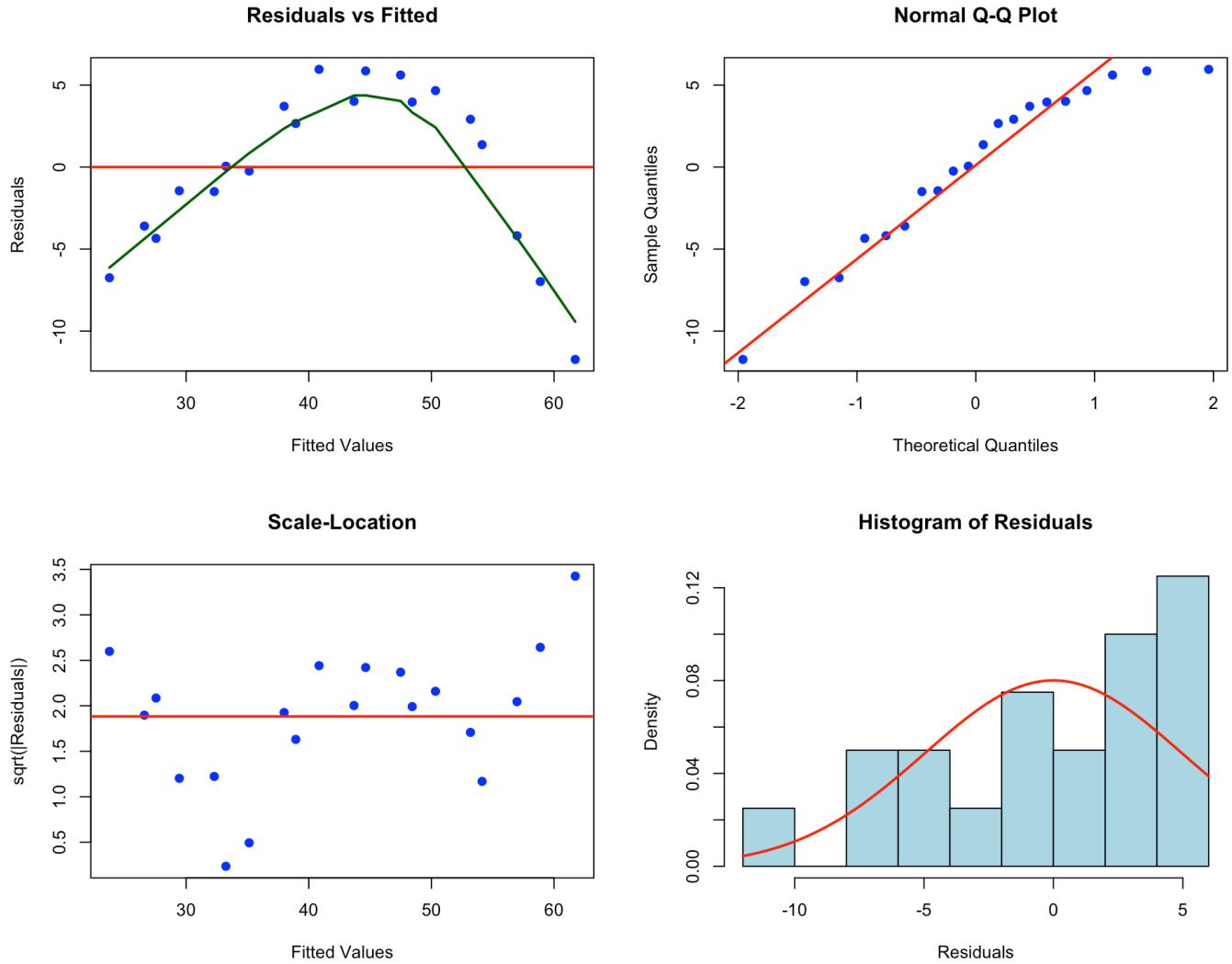
par(mfrow = c(2, 2))

# 1. Residuals vs Fitted
plot(fitted_values, residuals, main = "Residuals vs Fitted",
      xlab = "Fitted Values", ylab = "Residuals", pch = 19, col = "blue")
abline(h = 0, col = "red", lwd = 2)
lines(lowess(fitted_values, residuals), col = "darkgreen", lwd = 2)

# 2. Q-Q Plot
qqnorm(residuals, pch = 19, col = "blue")
qqline(residuals, col = "red", lwd = 2)

# 3. Scale-Location
plot(fitted_values, sqrt(abs(residuals)), main = "Scale-Location",
      xlab = "Fitted Values", ylab = "sqrt(|Residuals|)", pch = 19, col = "blue")
abline(h = mean(sqrt(abs(residuals))), col = "red", lwd = 2)

# 4. Histogram
hist(residuals, breaks = 10, probability = TRUE,
      main = "Histogram of Residuals", xlab = "Residuals", col = "lightblue")
curve(dnorm(x, mean(residuals), sd(residuals)), add = TRUE, col = "red", lwd = 2)
```

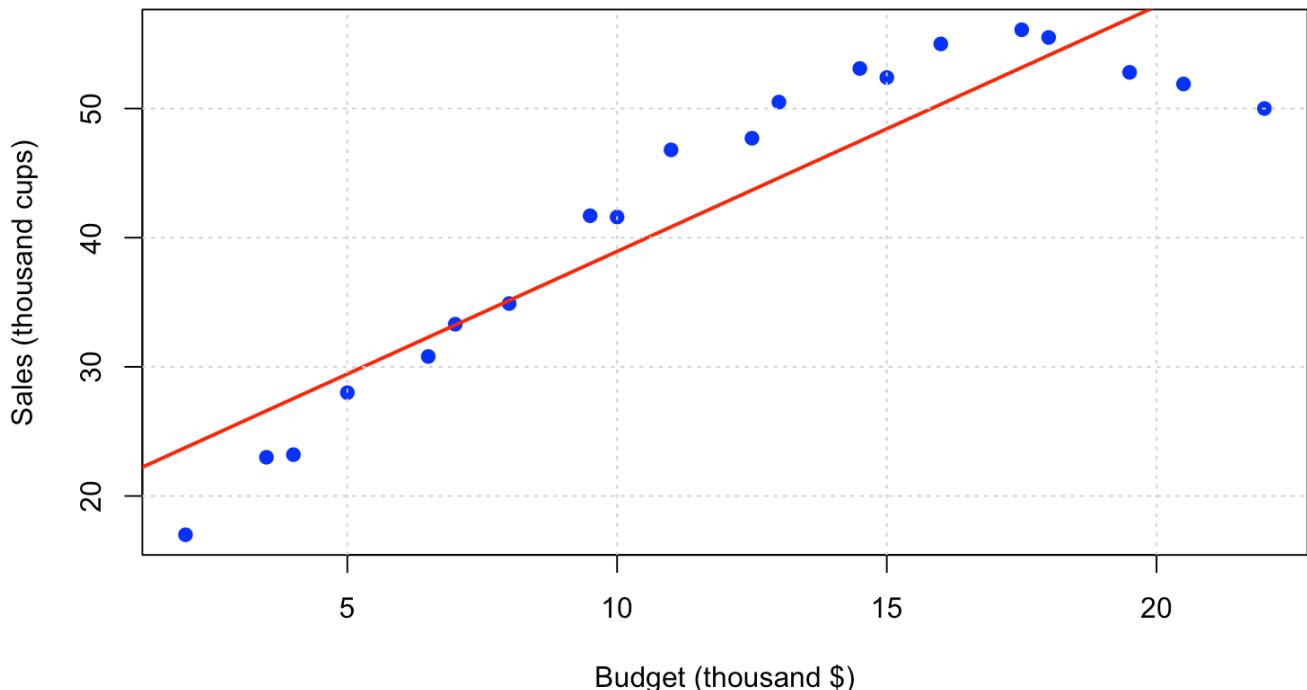


```

plot(x, y, pch = 19, col = "blue", main = "Advertising vs Sales with Regression Line",
      xlab = "Budget (thousand $)", ylab = "Sales (thousand cups)")
abline(model, col = "red", lwd = 2)
grid()

```

Advertising vs Sales with Regression Line



Assessment:

4.2.1 1. Normality of Residuals

```
shapiro.test(residuals)  
  
##  
## Shapiro-Wilk normality test  
##  
## data: residuals  
## W = 0.92735, p-value = 0.1373
```

Shapiro-Wilk: $W = 0.927$, $p = 0.137 > 0.05$. No evidence against normality.

Q-Q Plot: Points follow the line reasonably well with minor tail deviations.

Conclusion: ✓ Normality assumption satisfied

4.2.2 2. Linearity

Scatterplot: Shows strong linear trend from $x = 2$ to 16, but flattens/declines at higher budgets ($x > 16$).

Residuals vs Fitted: Lowess line shows curved U-shaped pattern, suggesting non-linearity. Residuals are positive at extremes, negative in middle.

Conclusion: Linearity questionable - Relationship may be non-linear (quadratic/plateau effect)

4.2.3 3. Homoscedasticity

Residuals vs Fitted: Vertical spread appears roughly constant (-12 to +6).

Scale-Location: No systematic fanning pattern.

Conclusion: ✓ Homoscedasticity reasonably satisfied

Overall: Model fits well ($R^2 = 0.84$) but the linearity violation is a concern. The U-shaped residual pattern indicates the simple linear model may not be appropriate for the full range of advertising budgets. This means:

- Predictions may be unreliable, particularly at higher advertising levels ($x > 16$) where the relationship flattens/declines
- The estimated slope ($\beta_1 = 1.90$) represents an average effect that doesn't capture the diminishing returns at higher budgets
- A more complex model (quadratic or segmented regression) would better capture the true relationship and provide more accurate predictions

Despite the high R^2 , caution is warranted when using this model for decision-making about advertising budgets beyond \$16,000.

4.3 (c) Other Variables

Two variables that could affect weekly coffee sales:

4.3.1 1. Weather/Temperature

- **Rationale:** Coffee consumption increases in colder weather; seasonal variations affect demand
- **Measurement:** Average weekly temperature ($^{\circ}\text{C}$), weather conditions (rainy/sunny)
- **Expected effect:** Negative correlation (colder \rightarrow higher sales)

4.3.2 2. Day of Week / Holidays

- **Rationale:** Weekday vs. weekend patterns differ; holidays and special events affect foot traffic
- **Measurement:** Number of weekdays in week, holiday indicator, special events count
- **Expected effect:** Varies (more weekdays near offices might increase sales)

Additional considerations: Competitor activity, economic indicators, product launches, store operations

Impact: Including these would improve prediction accuracy and help isolate true advertising effect from confounding factors.

5 Summary

All four questions have been completed with detailed solutions, proper statistical methodology, R code, and interpretations. Key findings:

- **Q1:** Die A has expected 45 successes ($SE = 4.47$); Die B significantly differs from Die A
 - **Q2:** Caffeine reduces reaction time by 5.31 ms on average ($p < 0.001$)
 - **Q3:** Alert athletes show significantly greater caffeine effect (7.40 ms vs 1.83 ms, $p < 0.001$)
 - **Q4:** Strong linear relationship between advertising and sales ($R^2 = 0.84$), though non-linearity evident at high budgets
-