SDA Group Submission Assignment Assign4

Group Gr18

MengliFeng (2720589) and PepijnVanOostveen (2801582)

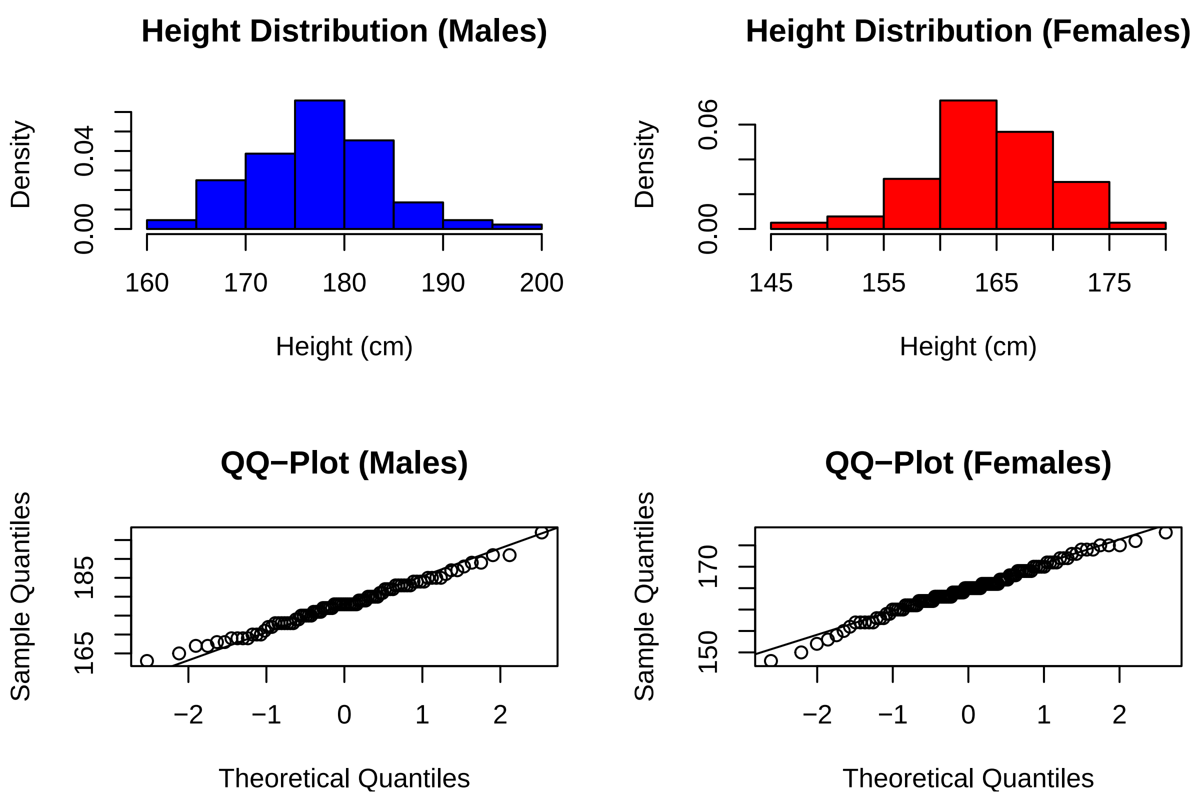
Exercise 1

```r
library(carData)
Davis <- Davis[-12, ]
head(Davis)
```

```
##   sex weight height repwt repht
## 1   M     77    182    77   180
## 2   F     58    161    51   159
## 3   F     53    161    54   158
## 4   M     68    177    70   175
## 5   F     59    157    59   155
## 6   M     76    170    76   165
```

a.

```r
# Load necessary libraries
library(ggplot2)

# Split data by sex
Davis_M <- subset(Davis, sex == "M")
Davis_F <- subset(Davis, sex == "F")

# Set up plotting area
par(mfrow = c(2, 2))

# Histograms for height distributions (scaled to density)
hist(Davis_M$height, probability = TRUE, main = "Height Distribution (Males)", xlab =
↪  "Height (cm)", col = "blue")
hist(Davis_F$height, probability = TRUE, main = "Height Distribution (Females)", xlab =
↪  "Height (cm)", col = "red")

# QQ-plots for normality check
qqnorm(Davis_M$height, main = "QQ-Plot (Males)")
qqline(Davis_M$height)

qqnorm(Davis_F$height, main = "QQ-Plot (Females)")
qqline(Davis_F$height)
```



```
# Normality test using Shapiro-Wilk
shapiro.test(Davis_M$height)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Davis_M$height
## W = 0.99196, p-value = 0.872
```

```
shapiro.test(Davis_F$height)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Davis_F$height
## W = 0.98947, p-value = 0.5468
```

from both the histograms and the qq-plots, we can see the distributions of heights of both male and female are both close to normal distribution. There are some wiggles observed in qq-plots, but they are likely from rounding effect, not necessarily linked to unnormality. From the histograms, we can observe a subtlety that heights of males have more mass below the mean and vice versa for weights of females.

From Shapiro-Wilk test results, the null hypothesis is not rejected.

b.

```
# Set seed for reproducibility
set.seed(1234)

# Function to compute the difference in means
```

```r
mean_diff <- function(data) {
  mean(data$height[data$sex == "M"]) - mean(data$height[data$sex == "F"])
}

# **Empirical Bootstrap**
n_bootstrap <- 1000  # Number of bootstrap samples
boot_diffs_empirical <- replicate(n_bootstrap, {
  resample_M <- sample(Davis_M$height, replace = TRUE)
  resample_F <- sample(Davis_F$height, replace = TRUE)
  mean(resample_M) - mean(resample_F)
})

# Standard deviation of bootstrap samples (empirical bootstrap)
sd_empirical <- sd(boot_diffs_empirical)

# **Parametric Bootstrap (assuming normality)**
parametric_boot_diffs <- replicate(n_bootstrap, {
  resample_M <- rnorm(length(Davis_M$height), mean(Davis_M$height), sd(Davis_M$height))
  resample_F <- rnorm(length(Davis_F$height), mean(Davis_F$height), sd(Davis_F$height))
  mean(resample_M) - mean(resample_F)
})

# Standard deviation of bootstrap samples (parametric bootstrap)
sd_parametric <- sd(parametric_boot_diffs)

# Print results
cat("Empirical Bootstrap SD:", sd_empirical, "\n")
```

```
## Empirical Bootstrap SD: 0.8418963
```

```r
cat("Parametric Bootstrap SD:", sd_parametric, "\n")
```

```
## Parametric Bootstrap SD: 0.8904208
```

c.

```r
# Given values
mu_M <- 177.8
mu_F <- 164.6
sigma_M <- 6.5
sigma_F <- 5.5
n_M <- length(Davis_M$height)
n_F <- length(Davis_F$height)

# Compute theoretical standard deviation
theoretical_sd <- sqrt((sigma_M^2 / n_M) + (sigma_F^2 / n_F))
cat("Theoretical SD:", theoretical_sd, "\n")
```

```
## Theoretical SD: 0.8675461
```

```r
# Compare theoretical vs. bootstrap estimates
comparison <- data.frame(
  Method = c("Empirical Bootstrap", "Parametric Bootstrap", "Theoretical"),
  SD = c(sd_empirical, sd_parametric, theoretical_sd)
)

print(comparison)
```

```
##                 Method        SD
## 1  Empirical Bootstrap 0.8418963
## 2 Parametric Bootstrap 0.8904208
## 3          Theoretical 0.8675461
```

the parametric boostrap estimation of the sd of the mean difference is closer to the theoretical value

d.

```
# Set boundaries for uniform distribution
min_M <- min(Davis_M$height)
max_M <- max(Davis_M$height)
min_F <- min(Davis_F$height)
max_F <- max(Davis_F$height)

# Perform parametric bootstrap with uniform distribution
uniform_boot_diffs <- replicate(n_bootstrap, {
  resample_M <- runif(n_M, min_M, max_M)
  resample_F <- runif(n_F, min_F, max_F)
  mean(resample_M) - mean(resample_F)
})

# Standard deviation of bootstrap samples (uniform distribution)
sd_uniform <- sd(uniform_boot_diffs)

cat("Uniform Bootstrap SD:", sd_uniform, "\n")
```

```
## Uniform Bootstrap SD: 1.313566
```

```
# Compare with theoretical standard deviation
comparison <- rbind(comparison, c("Uniform Bootstrap", sd_uniform))
print(comparison)
```

```
##                 Method                SD
## 1  Empirical Bootstrap 0.841896285568105
## 2 Parametric Bootstrap 0.890420792940573
## 3          Theoretical 0.867546055772349
## 4    Uniform Bootstrap  1.31356582282602
```

Exercise 2

a.

It is given that

$$\tilde{D_n} = max_i max\left(\left|\frac{i-1}{n} - \Phi\left(\frac{X_{(i)}-\bar{X}}{S}\right)\right|, \left|\frac{i}{n} - \Phi\left(\frac{X_{(i)}-\bar{X}}{S}\right)\right|\right)$$

This depends on F by using $\frac{X_{(i)}-\bar{X}}{S}$ and the null hypothesis states that $X_{(i)} \sim N(\bar{X}, S)$. Thus $X_{(i)} - \bar{X} \sim$

$N(0,S)$ and finally we define $Z_{(i)} = \frac{X_{(i)}-\bar{X}}{S} \sim N(0,1)$. Substituting this in our original equation we get:

$$\tilde{D_n} = max_i max\left(\left|\frac{i-1}{n} - \Phi\left(Z_{(i)}\right)\right|, \left|\frac{i}{n} - \Phi\left(Z_{(i)}\right)\right|\right)$$

Which shows that under the null, $\tilde{D}_n$ is independent of the location and scale parameters since $Z_{(i)}$ is independent of them.

b.

```r
# Load the data
data(morley)
head(morley)
```

```
##     Expt Run Speed
## 001    1   1   850
## 002    1   2   740
## 003    1   3   900
## 004    1   4  1070
## 005    1   5   930
## 006    1   6   850
```

```r
l_speed <- morley$Speed

# Calculate the mean and standard dev
m_speed <- mean(l_speed)
sd_speed <- sd(l_speed)

# run ks.test
ks_result <- ks.test(l_speed, "pnorm", mean = m_speed, sd = sd_speed)
```

```
## Warning in ks.test.default(l_speed, "pnorm", mean = m_speed, sd = sd_speed):
## ties should not be present for the one-sample Kolmogorov-Smirnov test
```

```r
# Extract D_n and the p-value
D_n <- ks_result$statistic
p_value <- ks_result$p.value
print(D_n)
```

```
##          D
## 0.08342437
```

```r
print(p_value)
```

```
## [1] 0.4895616
```

The p-value (0.4895616) is more than the significance level of 0.05 and thus we don't reject the null hypothesis that the data follows a normal distribution. The p-value isn't reliable since the Kolmogorov-Smirnov assumes that we now the parameters of the distribution, but we had to estimate them using the sample mean and standard deviation. (R also gives the warning "Warning: ties should not be present for the one-sample Kolmogorov-Smirnov test" which means that some values in the data repeat which and that shouldn't happen since the test is made for a continuous distribution.)

c.

```r
set.seed(1234)

n <- length(l_speed)
B <- 1000
Dn_star <- numeric(B)

for (i in 1:B) {
  # Generate bootstrap sample from N(m_speed, sd_speed^2)
  bootstrap_sample <- rnorm(n, mean = m_speed, sd = sd_speed)

  # Calculate the mean and standard dev
  m_star <- mean(bootstrap_sample)
  sd_star <- sd(bootstrap_sample)

  # run ks.test
  ks_star <- ks.test(bootstrap_sample, "pnorm", mean = m_star, sd = sd_star)
  Dn_star[i] <- ks_star$statistic
}

# compute p-value using the bootstrap distribution
p_value_bootstrap <- mean(Dn_star >= D_n)

print(p_value_bootstrap)
```

```
## [1] 0.078
```

d.

There is a big difference between the p-values in b and c. b results in a p-value of 0.4895616 and the bootstrap method in c results in a p-value of 0.078. This difference of 0.411 is very big, but still not large enough to result in different test decisions as the result from c is still above the significance level of 0.05. The final conclusion is that both tests result in not rejecting the null hypothesis and thus not having enough evidence that the data doesn't follow a normal distribution.