102b_hw02_Daren_Sathasivam

Daren Sathasivam

2024-04-14

Problem 1: Recall Problem 1 in Homework 1

Use the code from Homework 1 to simulate data from a bivariate normal distribution with mean vector $\mu = [0, 0]$ and correlation matrix $R = \begin{bmatrix} 1 & r \\ r & 1 \end{bmatrix}$

1. Sample size $n\epsilon\{50, 200\}$ and correlation coefficient r = 0

```
library(MASS)
set.seed(1)
mu <- c(0, 0)
sample_size <- c(50, 200)
# cor_coef <- c(0, 0.5, 0.85)

sim_data <- function(r, mu, sample_size) {
    for (n in sample_size) {
        Sigma <- matrix(c(1, r, r, 1), 2, 2)
        data <- mvrnorm(n = n, mu = mu, Sigma = Sigma)
        corr_coef <- cor(data[, 1], data[, 2])
        cat("Sample size: ", n, "Correlation coefficient: ",
            r, "Empirical Correlation: ", corr_coef, "\n")
        print(tail(data, 5))  # Print tail to show that the sample size is correct
    }
}

# Set correlation coefficient to 0
r <- 0
sim_data(r, mu, sample_size)
```

```
## Sample size:  50 Correlation coefficient:  0 Empirical Correlation:  0.03908718
##             [,1]       [,2]
## [46,] -0.5584864 -0.7074952
## [47,]  1.2765922  0.3645820
## [48,]  0.5732654  0.7685329
## [49,]  1.2246126 -0.1123462
## [50,]  0.4734006  0.8811077
## Sample size:  200 Correlation coefficient:  0 Empirical Correlation:  0.01046841
##              [,1]       [,2]
## [196,]  1.1089100  0.1344477
## [197,] -0.3075666  0.7655990
## [198,]  1.1068945  0.9551367
## [199,] -0.3476536 -0.0505657
```

```
## [200,]  0.8732645 -0.3058154
```

2. Sample size $n\epsilon\{50, 200\}$ and correlation coefficient r = 0.5

```
set.seed(1)
# Set correlation coefficient to 0.5
r <- 0.5
sim_data(r, mu, sample_size)
```

```
## Sample size:  50 Correlation coefficient:  0.5 Empirical Correlation:  0.3770455
##             [,1]       [,2]
## [46,] -0.8919520 -0.3334656
## [47,]  0.9540333 -0.3225589
## [48,]  0.9522017  0.3789363
## [49,]  0.5150116 -0.7096010
## [50,]  0.9997620  0.5263614
## Sample size:  200 Correlation coefficient:  0.5 Empirical Correlation:  0.4340609
##              [,1]       [,2]
## [196,]  0.6708901 -0.4380199
## [197,]  0.5092449  0.8168115
## [198,]  1.3806199  0.2737254
## [199,] -0.2176180  0.1300356
## [200,]  0.1717883 -0.7014762
```

3. Sample size $n\epsilon\{50, 200\}$ and correlation coefficient r = 0.85

```
set.seed(1)
# Set correlation coefficient to 0.85
r <- 0.85
sim_data(r, mu, sample_size)
```

```
## Sample size:  50 Correlation coefficient:  0.85 Empirical Correlation:  0.8018536
##             [,1]         [,2]
## [46,] -0.8333949 -0.527499247
## [47,]  0.7002529  0.001034529
## [48,]  0.8961465  0.582156099
## [49,]  0.2273228 -0.443425104
## [50,]  0.9770684  0.717776172
## Sample size:  200 Correlation coefficient:  0.85 Empirical Correlation:  0.8248348
##               [,1]        [,2]
## [196,]  0.43299513 -0.17437989
## [197,]  0.65209895  0.82056013
## [198,]  1.22175658  0.61548550
## [199,] -0.14384141  0.04657634
## [200,] -0.05497051 -0.53327719
```

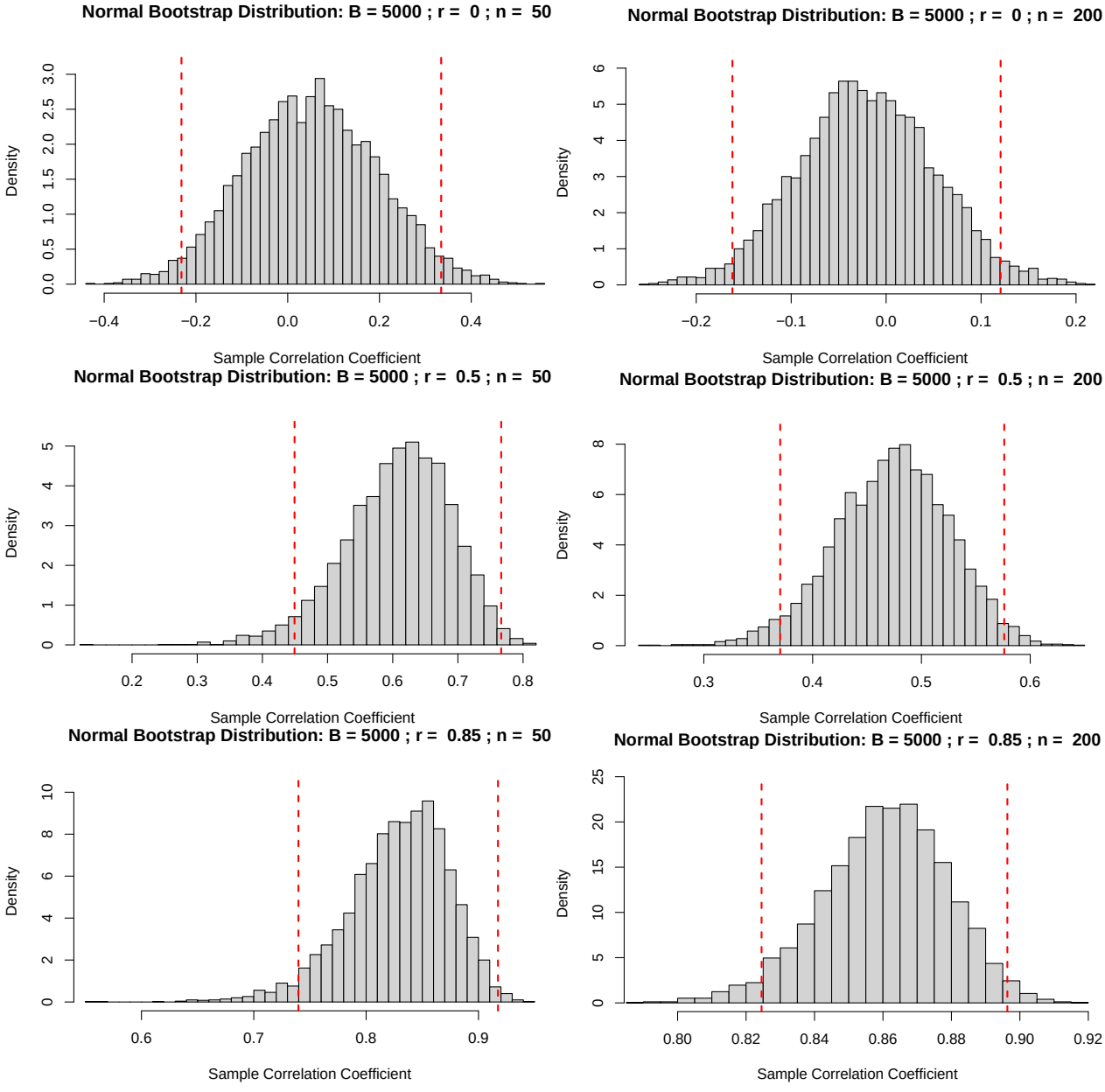
Obtain the following Bootstrap Confidence Intervals for the correlation coefficient r for the three cases:

1. Normal Bootstrap CI

```
# Load in helper functions provided from bruinlearn
source("bootsample.R")
source("bootstats.R")
```

```r
set.seed(1)
mu <- c(0, 0)
sample_size <- c(50, 200)
correlation_coeff <- c(0, 0.5, 0.85)
B <- 5000

# Function to calculate the correlation coefficient
cor_fun <- function(data) {
    cor(data[, 1], data[, 2])
}
# Function to perform bootstrap and calculate Normal CI
bs_normal_ci <- function(r, n, B) {
    Sigma <- matrix(c(1, r, r, 1), 2, 2)
    data <- mvrnorm(n = n, mu = mu, Sigma = Sigma)

    # Perform the bootstrap using custom bootsampling
    # function
    bootstrap_samples <- bootsampling(data, B)
    # Calculate statistics using custom boot.stats function
    boot_results <- boot.stats(bootstrap_samples, cor_fun)
    # Calculate Normal CI
    alpha <- 0.05
    zval <- qnorm(1 - alpha/2)  # Z-value for 95% CI
    boot_mean <- mean(boot_results$theta)
    boot_se <- boot_results$se

    normal_ci <- c(boot_mean - zval * boot_se, boot_mean + zval *
        boot_se)
    lower_ci <- normal_ci[1]
    upper_ci <- normal_ci[2]
    # Adjust histogram limits dynamically
    plot_range <- range(boot_results$theta)
    xlim_adjusted <- mean(plot_range) + c(-1, 1) * diff(plot_range)/2
    hist_data <- hist(boot_results$theta, breaks = 35, plot = FALSE)
    ylim_max <- max(hist_data$density) * 1.1

    # Plot histogram
    hist(boot_results$theta, breaks = 35, freq = FALSE, xlim = xlim_adjusted,
        ylim = c(0, ylim_max), main = paste("Normal Bootstrap Distribution: B =",
            B, "; r = ", r, "; n = ", n), xlab = "Sample Correlation Coefficient",
        ylab = "Density")
    abline(v = c(lower_ci, upper_ci), col = "red", lwd = 2, lty = 2)  # Add CI lines
    return(normal_ci)
}

# Run CI calculations for each scenario
for (r in correlation_coeff) {
    for (n in sample_size) {
        normal_ci <- bs_normal_ci(r, n, B)
        print(paste("Normal CIs for r = ", r, "and n =", n, ":"))
        print(normal_ci)
    }
}
```

```
## [1] "Normal CIs for r =  0 and n = 50 :"
## [1] -0.2315505  0.3345018

## [1] "Normal CIs for r =  0 and n = 200 :"
## [1] -0.1618761  0.1205628

## [1] "Normal CIs for r =  0.5 and n = 50 :"
## [1] 0.4493310 0.7664653

## [1] "Normal CIs for r =  0.5 and n = 200 :"
## [1] 0.3701688 0.5761160

## [1] "Normal CIs for r =  0.85 and n = 50 :"
## [1] 0.7397058 0.9173363

## [1] "Normal CIs for r =  0.85 and n = 200 :"
## [1] 0.8245289 0.8963461
```



2. Basic Bootstrap CI

```
bs_basic_ci <- function(r, n, B) {
    Sigma <- matrix(c(1, r, r, 1), 2, 2)
    data <- mvrnorm(n = n, mu = mu, Sigma = Sigma)

    # Perform bootstrap
    bootstrap_samples <- bootsampling(data, B)
    boot_results <- boot.stats(bootstrap_samples, cor_fun)
    # Calculate Basic CI
    alpha <- 0.05
    original_estimate <- cor_fun(data)
    lower_ci <- 2 * original_estimate - quantile(boot_results$theta,
        probs = 1 - alpha/2)
    upper_ci <- 2 * original_estimate - quantile(boot_results$theta,
        probs = alpha/2)

    # Adjust histogram limits dynamically
    plot_range <- range(boot_results$theta)
    xlim_adjusted <- mean(plot_range) + c(-1, 1) * diff(plot_range)/2
    hist_data <- hist(boot_results$theta, breaks = 35, plot = FALSE)
    ylim_max <- max(hist_data$density) * 1.1

    # Plot histogram
    hist(boot_results$theta, breaks = 35, freq = FALSE, xlim = xlim_adjusted,
        ylim = c(0, ylim_max), main = paste("Basic Bootstrap Distribution: B =",
            B, "; r = ", r, "; n = ", n), xlab = "Sample Correlation Coefficient",
        ylab = "Density")
    abline(v = c(lower_ci, upper_ci), col = "red", lwd = 2, lty = 2)  # Add CI lines
    return(c(lower_ci, upper_ci))
}

# Run CI calculations for each scenario
for (r in correlation_coeff) {
    for (n in sample_size) {
        basic_ci <- bs_basic_ci(r, n, B)
        print(paste("Basic CIs for r = ", r, "and n =", n, ":"))
        print(basic_ci)
    }
}
```

```
## [1] "Basic CIs for r =  0 and n = 50 :"
##       97.5%        2.5%
## -0.46087992 -0.08641575
```

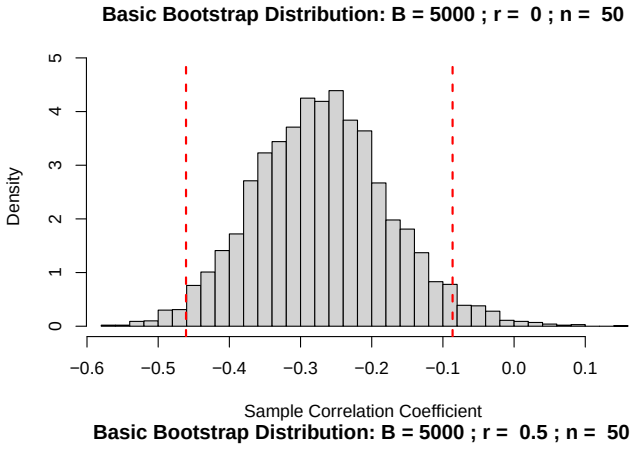
```
## [1] "Basic CIs for r =  0 and n = 200 :"
##      97.5%       2.5%
## -0.1346108  0.1569481
```

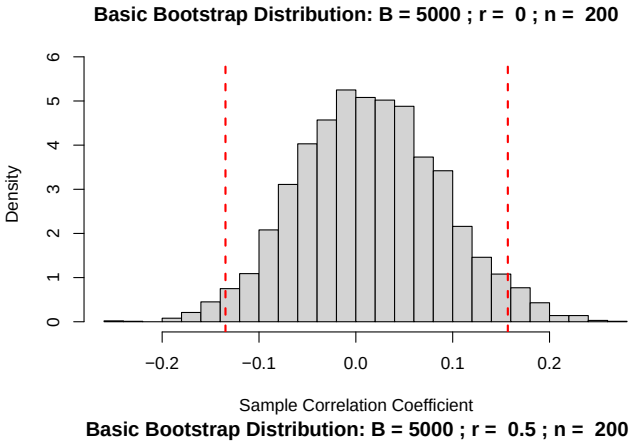
```
## [1] "Basic CIs for r =  0.5 and n = 50 :"
##     97.5%      2.5%
## 0.2169694 0.7452679
```

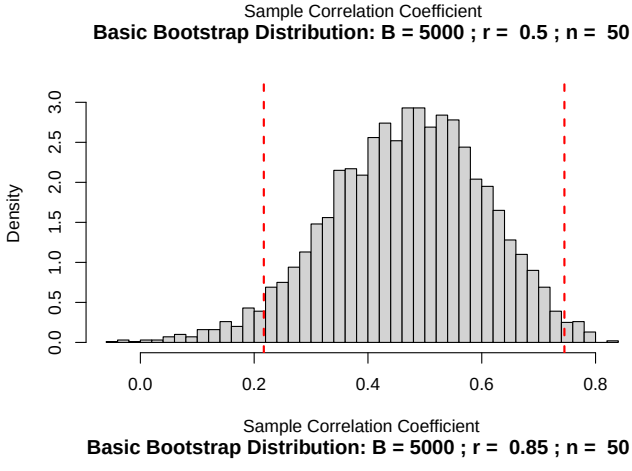
```
## [1] "Basic CIs for r =  0.5 and n = 200 :"
##     97.5%      2.5%
## 0.4220135 0.6284923
```

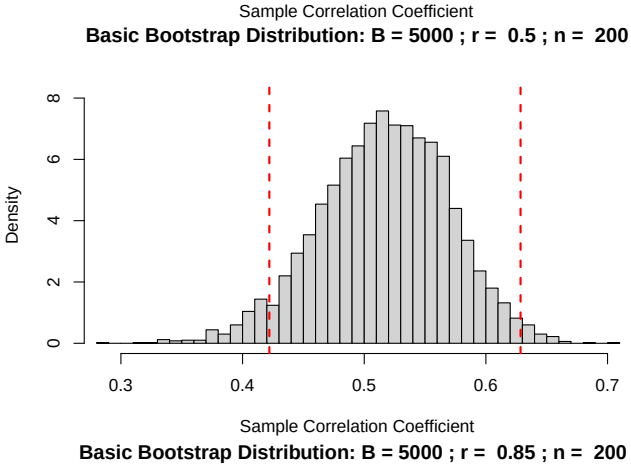
```
## [1] "Basic CIs for r =  0.85 and n = 50 :"
##     97.5%      2.5%
## 0.8341367 0.9723643
```

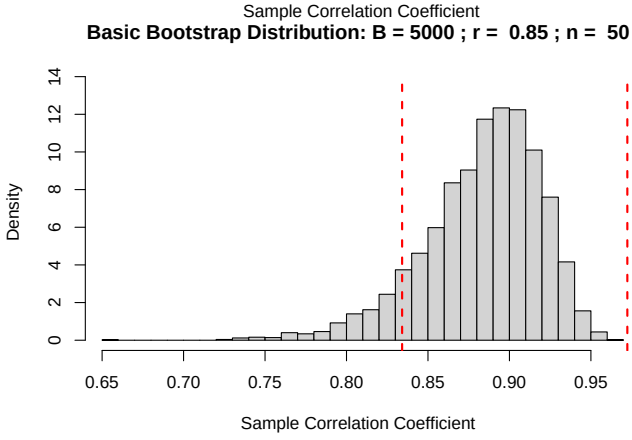
```
## [1] "Basic CIs for r =  0.85 and n = 200 :"
##     97.5%      2.5%
## 0.7857368 0.8740764
```

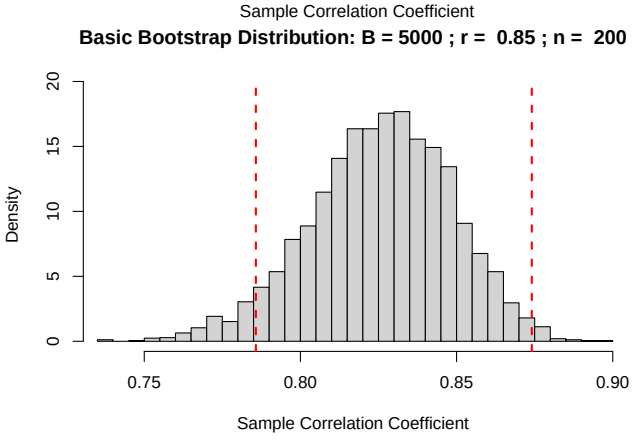












3. Precentile Bootstrap CI

```
bs_percentile_ci <- function(r, n, B) {
    Sigma <- matrix(c(1, r, r, 1), 2, 2)
    data <- mvrnorm(n = n, mu = mu, Sigma = Sigma)

    # Perform bootstrap
```

```
    bootstrap_samples <- bootsampling(data, B)
    boot_results <- boot.stats(bootstrap_samples, cor_fun)

    # Calculate Percentile CI
    alpha <- 0.05
    # percentile.CI=rbind(percentile.CI,cbind(quantile(x.mean$theta,probs=(alpha/2)),
    # quantile(x.mean$theta,probs=(1-alpha/2))))
    lower_ci <- quantile(boot_results$theta, probs = alpha/2)
    upper_ci <- quantile(boot_results$theta, probs = 1 - alpha/2)

    # Adjust histogram limits dynamically
    plot_range <- range(boot_results$theta)
    xlim_adjusted <- mean(plot_range) + c(-1, 1) * diff(plot_range)/2
    hist_data <- hist(boot_results$theta, breaks = 35, plot = FALSE)
    ylim_max <- max(hist_data$density) * 1.1

    # Plot histogram
    hist(boot_results$theta, breaks = 35, freq = FALSE, xlim = xlim_adjusted,
        ylim = c(0, ylim_max), main = paste("Percentile Bootstrap Distribution: B =",
            B, "; r = ", r, "; n = ", n), xlab = "Sample Correlation Coefficient",
        ylab = "Density")
    abline(v = c(lower_ci, upper_ci), col = "red", lwd = 2, lty = 2)  # Add CI lines
    return(c(lower_ci, upper_ci))
}

# Run CI calculations for each scenario
for (r in correlation_coeff) {
    for (n in sample_size) {
        percentile_ci <- bs_percentile_ci(r, n, B)
        print(paste("Percentile CIs for r = ", r, "and n =",
            n, ":"))
        print(percentile_ci)
    }
}
```

```
## [1] "Percentile CIs for r =  0 and n = 50 :"
##       2.5%      97.5%
## -0.3700901  0.2154169
```

```
## [1] "Percentile CIs for r =  0 and n = 200 :"
##       2.5%      97.5%
## -0.1751147  0.1539815
```

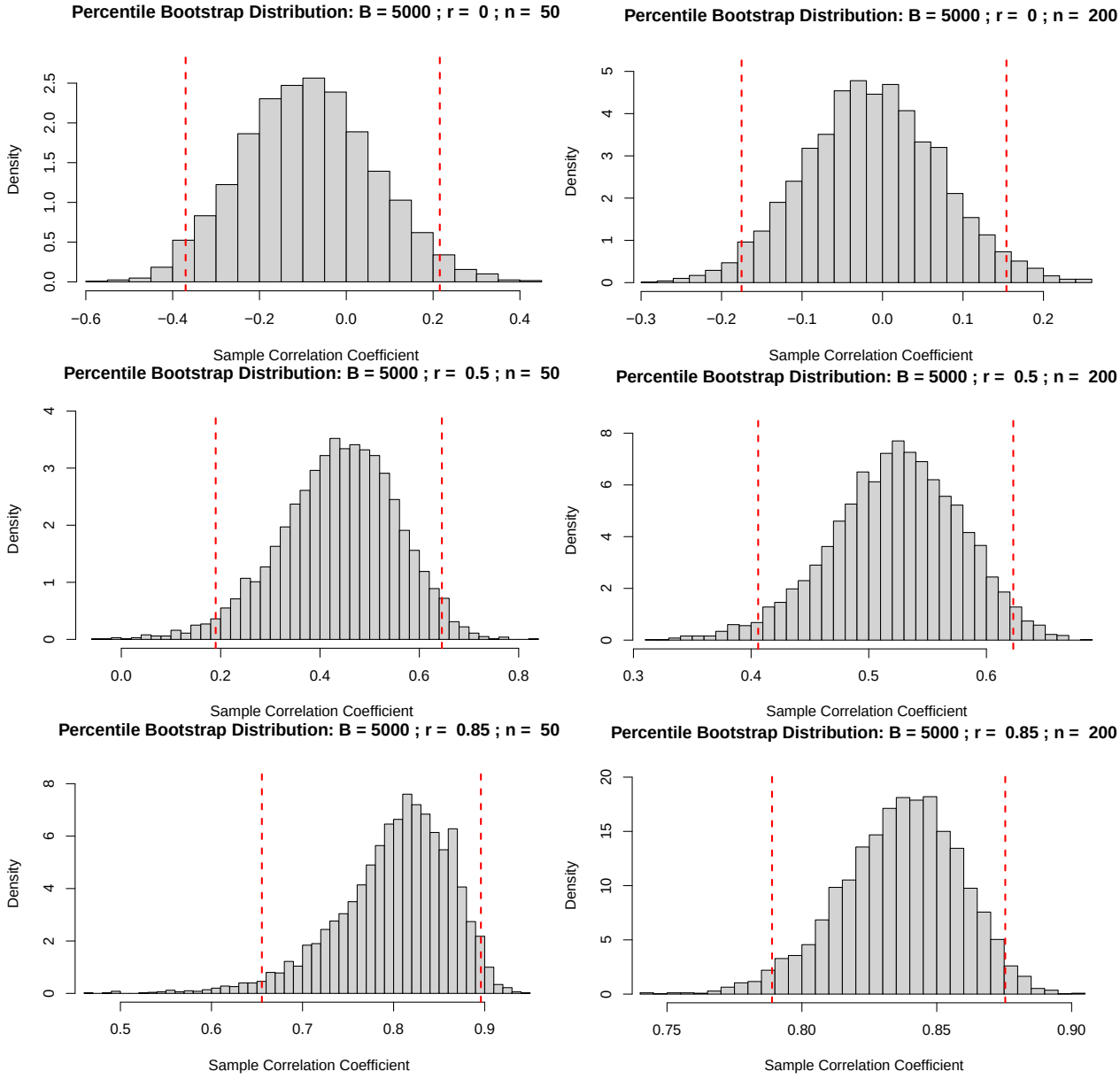
```
## [1] "Percentile CIs for r =  0.5 and n = 50 :"
##      2.5%     97.5%
## 0.1900326 0.6453949
```

```
## [1] "Percentile CIs for r =  0.5 and n = 200 :"
##      2.5%     97.5%
## 0.4060088 0.6228278
```

```
## [1] "Percentile CIs for r =  0.85 and n = 50 :"
##      2.5%     97.5%
## 0.6553675 0.8958566
```

```
## [1] "Percentile CIs for r =  0.85 and n = 200 :"
```

```
##      2.5%     97.5%
## 0.7889437 0.8753991
```



Calculate the length and the shape of each type of Bootstrap CI and report:

```
set.seed(1)
mu <- c(0, 0)
sample_size <- c(50, 200)
correlation_coeff <- c(0, 0.5, 0.85)
B <- 5000
pdf(NULL)  # So graphs don't display when running function
for (r in correlation_coeff) {
    for (n in sample_size) {
        # Calculate Normal CI and its length
        normal_ci <- bs_normal_ci(r, n, B)
```

```
        normal_length <- normal_ci[2] - normal_ci[1]
        normal_midpoint <- mean(normal_ci)

        # Calculate Basic CI and its length
        basic_ci <- bs_basic_ci(r, n, B)
        basic_length <- basic_ci[2] - basic_ci[1]
        basic_midpoint <- mean(basic_ci)

        # Calculate Percentile CI and its length
        percentile_ci <- bs_percentile_ci(r, n, B)
        percentile_length <- percentile_ci[2] - percentile_ci[1]
        percentile_midpoint <- mean(percentile_ci)

        # Print results
        cat(paste("For r =", r, "and n =", n, ":\n"))
        cat(paste("  Normal CI Length:", normal_length, "Midpoint:",
            normal_midpoint, "\n"))
        cat(paste("  Basic CI Length:", basic_length, "Midpoint:",
            basic_midpoint, "\n"))
        cat(paste("  Percentile CI Length:", percentile_length,
            "Midpoint:", percentile_midpoint, "\n\n"))
    }
}
```

```
## For r = 0 and n = 50 :
##   Normal CI Length: 0.56605226005074 Midpoint: 0.0514756494553296
##   Basic CI Length: 0.634705112417542 Midpoint: 0.00190363485366363
##   Percentile CI Length: 0.512409279826494 Midpoint: -0.038902443732848
```

```
## For r = 0 and n = 200 :
##   Normal CI Length: 0.268356009348293 Midpoint: 0.0506282408985696
##   Basic CI Length: 0.28087663640995 Midpoint: 0.035940117181502
##   Percentile CI Length: 0.269977338451089 Midpoint: -0.0240833904646406
```

```
## For r = 0.5 and n = 50 :
##   Normal CI Length: 0.420577817272784 Midpoint: 0.42389570609189
##   Basic CI Length: 0.4171517049659 Midpoint: 0.546578741748757
##   Percentile CI Length: 0.411305126151462 Midpoint: 0.515811354015365
```

```
## For r = 0.5 and n = 200 :
##   Normal CI Length: 0.179602316430254 Midpoint: 0.575652371522082
##   Basic CI Length: 0.229206911462323 Midpoint: 0.488779634104918
##   Percentile CI Length: 0.218984859013633 Midpoint: 0.440932849938248
```

```
## For r = 0.85 and n = 50 :
##   Normal CI Length: 0.188955262539505 Midpoint: 0.773811399416126
##   Basic CI Length: 0.117086078273916 Midpoint: 0.862941237906687
##   Percentile CI Length: 0.16745115391022 Midpoint: 0.824099620194776
```

```
## For r = 0.85 and n = 200 :
##   Normal CI Length: 0.0859287984316441 Midpoint: 0.82845674984908
##   Basic CI Length: 0.0776704665575174 Midpoint: 0.864042266748143
##   Percentile CI Length: 0.0632367685586753 Midpoint: 0.859724523393821
```

```
dev.off()
```

```
## pdf
```

```
##   2
```

- Normal Bootstrap CI: displays consistent midpoints close to the true correlation coefficient values, especially at higher sample sizes which indicates good adherence to the normality assumption in the bootstrap distribution.
- Basic Bootstrap CI: displays greater variability in its midpoints, reflecting its non-parametric nature and sensitivity to skewness which indicates the versatility but less stable nature.
- Percentile Bootstrap CI: displays midpoints that closely align with true values in larger samples, indicating it effectively captures the central tendency of the distribution with less sensisivity to outliers in comparison to the other methods.

Discuss how you selected the number of bootstrap replicates B:
Hint: The results for the population median in the Lectures of Week 3 provide good guidance onw hat is an appropriate B.
- I chose 5000 bootstrap replications as it strikes a good balance between efficiency and accuracy as seen by the output of various population medians in lecture. This value ensures that the confidence intervals and other derived statistics are both reliable and computationally feasible to calculate.

Comment on the results; in particular how the various bootstrap CI behave as a function of the sample size n, and the value of the correlation coefficient r.
- In relation to how the various bootstrap CI behave as a function of sample size n, it is observed that they affect confidence interval length and convergence. Larger sample sizes tend to provide more accurate results for the length of confidence interval as they provide more information and provide a convergence towards the true correlation coefficient. Additionally, the correlation coefficient r has an impact on the length of confidence interval as r = 0, the length is much broader whereas r = 0.85 provides a smaller confidence interval length.

Problem 2:

```r
library(MASS)
data(cats)
summary(cats)
```

```
##  Sex         Bwt             Hwt
##  F:47   Min.   :2.000   Min.   : 6.30
##  M:97   1st Qu.:2.300   1st Qu.: 8.95
##         Median :2.700   Median :10.10
##         Mean   :2.724   Mean   :10.63
##         3rd Qu.:3.025   3rd Qu.:12.12
##         Max.   :3.900   Max.   :20.50
```

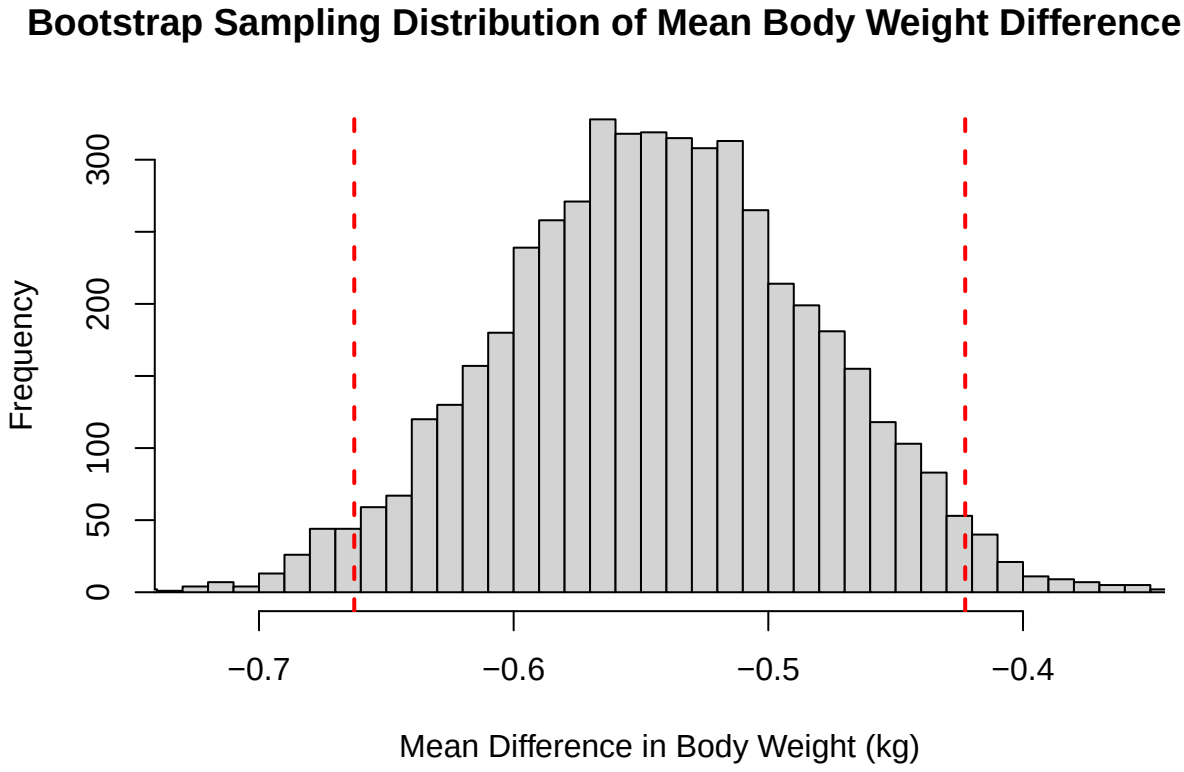
```r
# BWT in kg and HWT in g of 47F and 97M cats

# F/M Body Weight Data
female_bwt <- cats$Bwt[cats$Sex == "F"]  # 47 obs
male_bwt <- cats$Bwt[cats$Sex == "M"]  # 97 obs
# F/M Heart Weight Data
female_hwt <- cats$Hwt[cats$Sex == "F"]  # 47 obs
male_hwt <- cats$Hwt[cats$Sex == "M"]  # 97 obs
```

Part (a):

Construct the following bootstrap CI for the **difference of the body weight means** between female and male cats.

1. Normal Bootstrap CI

```r
set.seed(1)
# Number of bootstrap replicates
B <- 5000
# Function to perform bootstrap sampling and calculate mean
# differences
bootstrap_mean_diff <- function(f_data, m_data, B) {
    mean_diffs <- rep(NA, B)
    # Iterate through number of bootstrap replicates
    for (i in seq_along(mean_diffs)) {
        f_sample <- sample(f_data, replace = TRUE)
        m_sample <- sample(m_data, replace = TRUE)
        mean_diffs[i] <- mean(f_sample) - mean(m_sample)
    }
    mean_diffs
}
# Perform bootstrap sampling
bwt_diffs <- bootstrap_mean_diff(female_bwt, male_bwt, B)

# Calculate Normal Bootstrap CI
alpha <- 0.05
z_value <- qnorm(1 - alpha/2)
mean_diff <- mean(bwt_diffs)
sd_diff <- sd(bwt_diffs)
normal_lower <- mean_diff - z_value * sd_diff
normal_upper <- mean_diff + z_value * sd_diff
# Plot bootstrap sampling distribution
hist(bwt_diffs, breaks = 40, main = "Bootstrap Sampling Distribution of Mean Body Weight Difference (F-
    xlab = "Mean Difference in Body Weight (kg)", ylab = "Frequency",
    xlim = c(mean_diff - 3 * sd_diff, mean_diff + 3 * sd_diff))
abline(v = c(normal_lower, normal_upper), col = "red", lwd = 2,
    lty = 2)
```




```
# Print normal bootstrap CI results
cat("Normal Bootstrap CI for the difference in mean body weight (female - male): \n[",
    normal_lower, ",", normal_upper, "] \nNormal Bootstrap CI Length: ",
    normal_upper - normal_lower, "\n")
```

```
## Normal Bootstrap CI for the difference in mean body weight (female - male):
## [ -0.6625853 , -0.4226978 ]
## Normal Bootstrap CI Length:  0.2398875
```

2. Basic Bootstrap CI

```
# Calculate Basic Bootstrap CI:
alpha <- 0.05
z_value <- qnorm(1 - alpha/2)
mean_diff <- mean(bwt_diffs)
sd_diff <- sd(bwt_diffs)

basic_lower <- 2 * mean(bwt_diffs) - quantile(bwt_diffs, 1 -
    alpha/2)
basic_upper <- 2 * mean(bwt_diffs) - quantile(bwt_diffs, alpha/2)

cat("Basic Bootstrap CI for the difference in mean body weight (female - male): \n[",
    basic_lower, ",", basic_upper, "] \nBasic Bootstrap CI Length: ",
    basic_upper - basic_lower, "\n")
```

```
## Basic Bootstrap CI for the difference in mean body weight (female - male):
## [ -0.6599343 , -0.4190388 ]
## Basic Bootstrap CI Length:  0.2408955
```

3. Percentile Bootstrap CI

```
# Calculate Percentile Bootstrap CI:
alpha <- 0.05
z_value <- qnorm(1 - alpha/2)
mean_diff <- mean(bwt_diffs)
sd_diff <- sd(bwt_diffs)

percentile_lower <- quantile(bwt_diffs, alpha/2)
percentile_upper <- quantile(bwt_diffs, 1 - alpha/2)

cat("Percentile Bootstrap CI for the difference in mean body weight (female - male): \n[",
    percentile_lower, ",", percentile_upper, "] \nPercentile Bootstrap CI Length: ",
    percentile_upper - percentile_lower, "\n")
```

```
## Percentile Bootstrap CI for the difference in mean body weight (female - male):
## [ -0.6662442 , -0.4253488 ]
## Percentile Bootstrap CI Length:  0.2408955
```

Calculate the length and shape of each type of Bootstrap CI and report them as well.
- For Normal Bootstrap CI Length, we obtain 0.2398875 whereas Basic and Percentile obtain a length of 0.2408955. We can observe that all three bootstrap methods result in similar confidence interval lengths, indicating that choice of method may not significantly affect conclusions drawn from the data under these conditions. Similarly, all three methods reflect a similar distribution shape and variability.

Discuss how you selected the number of bootstrap replicates B:
- I chose 5000 bootstrap replicates as it offers a good balance between computational efficiency and accuracy in comparison to replicates that are too small or too large.

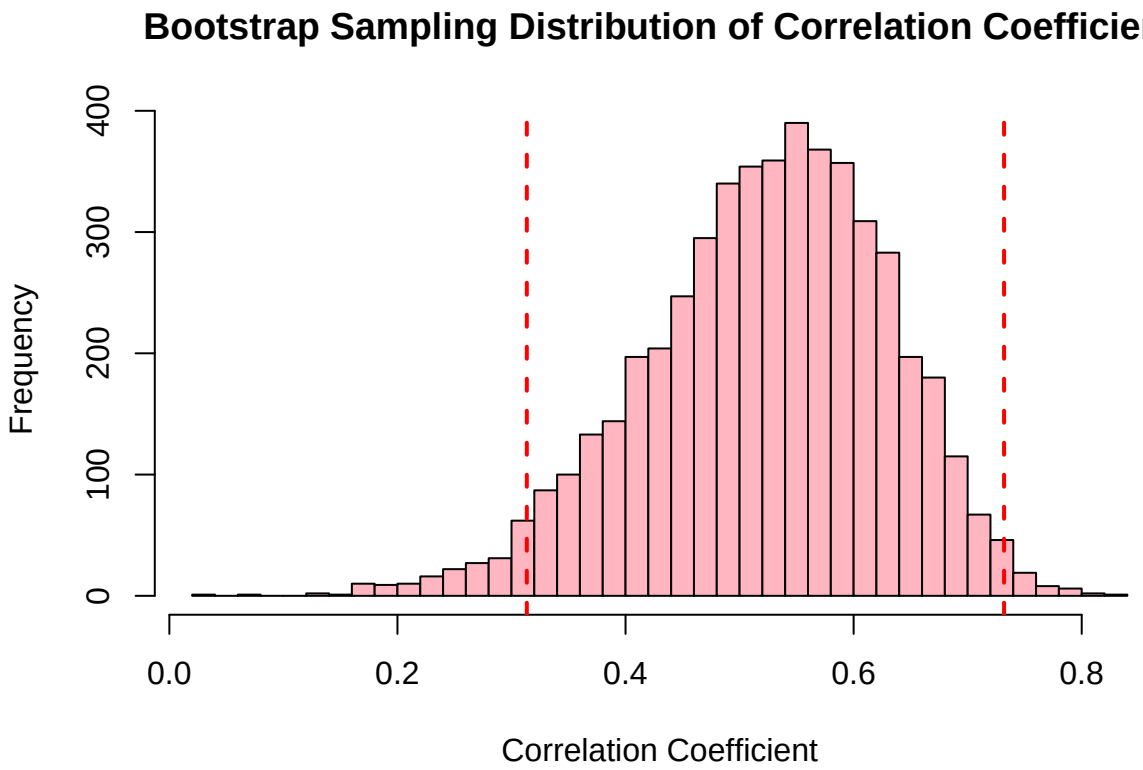
Part (b)

Using code from Problem 1 above to construct the following bootstrap CI for the **correlation coefficient between the body weight and heart weight** of female cats:

1. Normal Bootstrap CI

```
set.seed(1)
# Number of Bootstrap replicates
B <- 5000
# Data for all female cats
female_cats <- cats[cats$Sex == "F", ]
# Function to perform normal bootstrap CI
bs_normal_cor_ci <- function(data, B) {
    # Perform bootstrap
    bootstrap_results <- boot(data[, c("Bwt", "Hwt")], statistic = cor_fun2,
        R = B)
    # Calculate Normal CI
    alpha <- 0.05
    zval <- qnorm(1 - alpha/2)
    boot_mean <- mean(bootstrap_results$t)
    boot_sd <- sd(bootstrap_results$t)

    normal_ci <- c(boot_mean - zval * boot_sd, boot_mean + zval *
        boot_sd)
    return(normal_ci)
```

```
}
# Calculate Normal Bootstrap CI
normal_ci <- bs_normal_cor_ci(female_cats, B)

# Plot the histogram of bootstrap results
hist(bootstrap_results$t, breaks = 40, main = "Bootstrap Sampling Distribution of Correlation Coefficie
    xlab = "Correlation Coefficient", ylab = "Frequency", col = "lightpink")
abline(v = normal_ci[1], col = "red", lwd = 2, lty = 2)
abline(v = normal_ci[2], col = "red", lwd = 2, lty = 2)
```



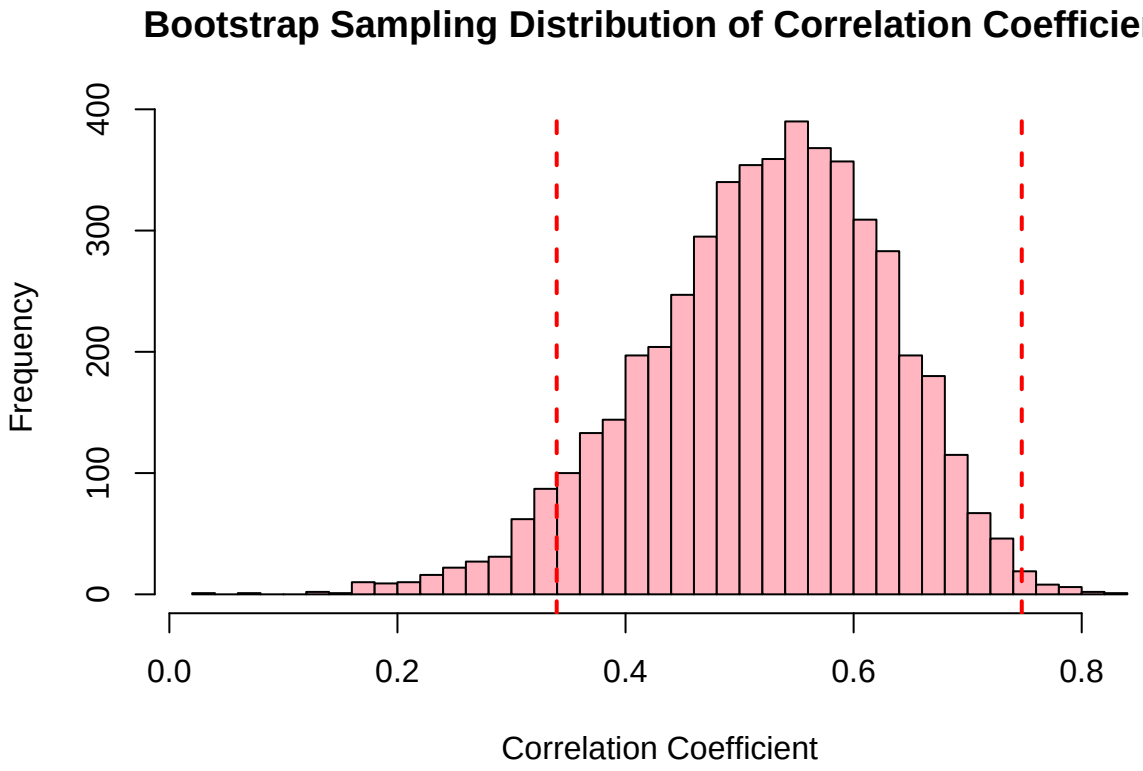

```
cat("Normal Bootstrap CI for the correlation between body weight and heart weight (female cats): \n[",
    normal_ci[1], ",", normal_ci[2], "] \nNormal Bootstrap CI Length: ",
    normal_ci[2] - normal_ci[1], "\n")
```

```
## Normal Bootstrap CI for the correlation between body weight and heart weight (female cats):
## [ 0.3134269 , 0.7319284 ]
## Normal Bootstrap CI Length:  0.4185015
```

2. Basic Bootstrap CI

```
set.seed(1)
# Perform the bootstrap for correlation coefficients
female_cats <- cats[cats$Sex == "F", c("Bwt", "Hwt")]
# Basic bootstrap CI for correlation
bs_basic_cor_ci <- function(data, B) {
    # Perform bootstrap
    bootstrap_results <- boot(data[, c("Bwt", "Hwt")], statistic = cor_fun2,
        R = B)
    # Calculate Basic CI Calculate Basic CI
```

```
    alpha <- 0.05
    original_estimate <- mean(bootstrap_results$t)  # Mean of bootstrap estimates as the original estim
    lower_ci <- 2 * original_estimate - quantile(bootstrap_results$t,
        probs = 1 - alpha/2)
    upper_ci <- 2 * original_estimate - quantile(bootstrap_results$t,
        probs = alpha/2)

    return(c(lower_ci, upper_ci))
}
# Calculate Basic Bootstrap CI
basic_ci <- bs_basic_cor_ci(female_cats, B)

# Plot the histogram of bootstrap results
hist(bootstrap_results$t, breaks = 40, main = "Bootstrap Sampling Distribution of Correlation Coefficie
    xlab = "Correlation Coefficient", ylab = "Frequency", col = "lightpink")
abline(v = basic_ci[1], col = "red", lwd = 2, lty = 2)
abline(v = basic_ci[2], col = "red", lwd = 2, lty = 2)
```



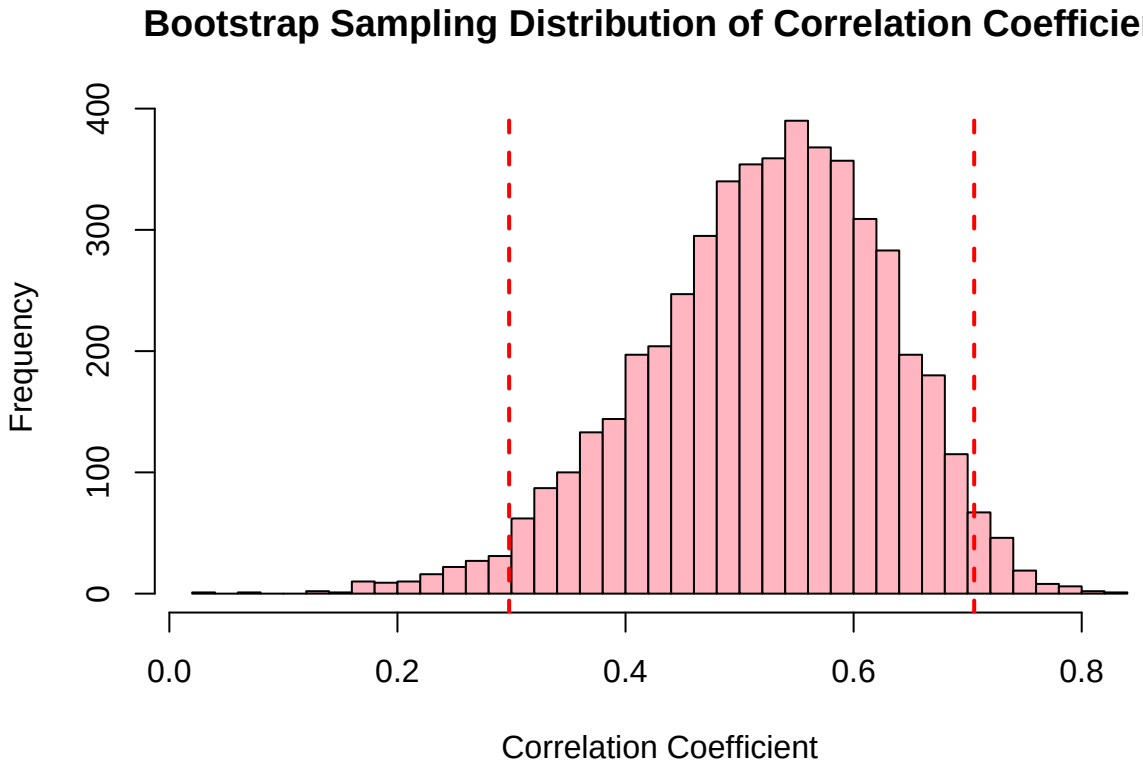

```
cat("Basic Bootstrap CI for the correlation between body weight and heart weight (female cats): \n[",
    basic_ci[1], ",", basic_ci[2], "] \nBasic Bootstrap CI Length: ",
    basic_ci[2] - basic_ci[1], "\n")
```

```
## Basic Bootstrap CI for the correlation between body weight and heart weight (female cats):
## [ 0.3396359 , 0.7474276 ]
## Basic Bootstrap CI Length:  0.4077918
```

3. Percentile Bootstrap CI

```
set.seed(1)
# Perform the bootstrap for correlation coefficients
```

```r
female_cats <- cats[cats$Sex == "F", c("Bwt", "Hwt")]
# Percentile bootstrap CI for correlation function
bs_percentile_cor_ci <- function(data, B) {
    # Perform bootstrap
    bootstrap_results <- boot(data, statistic = cor_fun2, R = B)
    # Calculate Percentile CI
    alpha <- 0.05
    lower_ci <- quantile(bootstrap_results$t, probs = alpha/2)
    upper_ci <- quantile(bootstrap_results$t, probs = 1 - alpha/2)
    return(c(lower_ci, upper_ci))
}
# Calculate Basic Bootstrap CI
percentile_ci <- bs_percentile_cor_ci(female_cats, B)

# Plot the histogram of bootstrap results
hist(bootstrap_results$t, breaks = 40, main = "Bootstrap Sampling Distribution of Correlation Coefficie
    xlab = "Correlation Coefficient", ylab = "Frequency", col = "lightpink")
abline(v = percentile_ci[1], col = "red", lwd = 2, lty = 2)
abline(v = percentile_ci[2], col = "red", lwd = 2, lty = 2)
```




```r
cat("Percentile Bootstrap CI for the correlation between body weight and heart weight (female cats): \n
    percentile_ci[1], ",", percentile_ci[2], "] \nPercentile Bootstrap CI Length: ",
    percentile_ci[2] - percentile_ci[1], "\n")
```

```
## Percentile Bootstrap CI for the correlation between body weight and heart weight (female cats):
## [ 0.2979277 , 0.7057195 ]
## Percentile Bootstrap CI Length:  0.4077918
```

Calculate their length and their shape and report those as well for each type of bootstrap CI.
- After calculating the bootstrap sampling distribution of the correlation between body weight and heart

weight for female cats, it can be observed that the length of the confidence intervals for all three methods fall roughly around 0.41. Each has a similar shape with a central tendency near 0.57 which can be observed by the histogram. Specifically, the length of Normal CI is 0.4185015, Basic CI is 0.4077918, and Percentile CI is 0.4077918.

Discuss how you selected the number of bootstrap replicates B and comment on the results.
- I selected 5000 as the number of bootstrap replicates so the bootstrap sampling distribution can display an accurate model while aso remaining efficient when computing.

Part (c)

Based on the results in Part(b), what are your conclusions regarding the following two statements:

```
# Get data
female_bwt <- cats$Bwt[cats$Sex == "F"]  # 47 obs
male_bwt <- cats$Bwt[cats$Sex == "M"]  # 97 obs
female_hwt <- cats$Hwt[cats$Sex == "F"]  # 47 obs
male_hwt <- cats$Hwt[cats$Sex == "M"]  # 97 obs
# Observed differences
obs_diff_bwt <- mean(female_bwt) - mean(male_bwt)
obs_diff_hwt <- mean(female_hwt) - mean(male_hwt)
# Perform bootstrap sampling (using problem 2a)
bwt_diffs <- bootstrap_mean_diff(female_bwt, male_bwt, B)
hwt_diffs <- bootstrap_mean_diff(female_hwt, male_hwt, B)

# Basic Bootstrap Sampling CI:
alpha <- 0.05
z_value <- qnorm(1 - alpha/2)
# Use basic bootstrap CI for BWT (use observed mean diff to
# calculate)
bwt_basic_lower <- 2 * obs_diff_bwt - quantile(bwt_diffs, 1 -
    alpha/2)
bwt_basic_upper <- 2 * obs_diff_bwt - quantile(bwt_diffs, alpha/2)
# For HWT
hwt_basic_lower <- 2 * obs_diff_hwt - quantile(hwt_diffs, 1 -
    alpha/2)
hwt_basic_upper <- 2 * obs_diff_hwt - quantile(hwt_diffs, alpha/2)

# Print results
cat("Bootstrap CI for difference in mean body weight between genders: [",
    bwt_basic_lower, ",", bwt_basic_upper, "]\n")
```

```
## Bootstrap CI for difference in mean body weight between genders: [ -0.6559788 , -0.4170361 ]
```

```
cat("Bootstrap CI for difference in mean heart weight between genders: [",
    hwt_basic_lower, ",", hwt_basic_upper, "]\n")
```

```
## Bootstrap CI for difference in mean heart weight between genders: [ -2.736527 , -1.498715 ]
```

- The body weight means of female and male cats are equal
  - Based off of the confidence interval for the difference in mean body weights between females and males([-0.6590524 , -0.420947]), the interval is entirely below zero. This indicates that the mean body weight of males is higher than that of female cats, thus the body weight means of female and male cats are NOT equal.
- The heart weight means of female and male cats are equal

– Similar to body weight, the difference in mean heart weight between females and males([-2.765632 , -1.467682]) is also entirely below zero. This also indicates that the mean heart weight of males is higher than females, thus the heart weight means of female and male cats are NOT equal.