CS-6313 – Statistical Methods for Data Science

Mini Project #5

Group No - 5
Manan Dalal (MUD200000)
Lipi Patel (LDP210000)

Contribution of Team Members:

We both collaborated and solved both questions together for a thorough understanding of functions in R and solved both questions concurrently to check for accuracy, debugging and application.

Question 1:

Consider the data stored in bodytemp-heartrate.csv on eLearning, containing measurements of body temperature and heart rate for 65 male (gender = 1) and 65 female (gender = 2) subjects.

Solution

⇨ First, we read the csv file and divide the data into male and female data.

```
# Reading data
heartRateData = read.csv("E:/MS-CS/Spring 22/CS6313 - SMDS/Mini Projects/5/bodytemp-heartrate.csv")

# Separting the database based on gender
maleData = subset(heartRateData, heartRateData$gender == 1)
femaleData = subset(heartRateData, heartRateData$gender == 2)
```

a. Do males and females differ in mean body temperature? Answer this question by performing an appropriate analysis of the data, including an exploratory analysis.

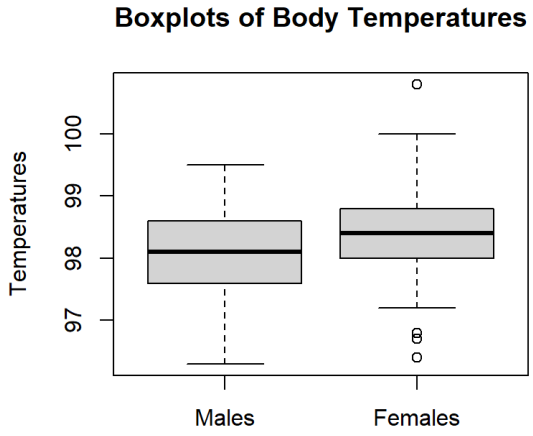
Solution

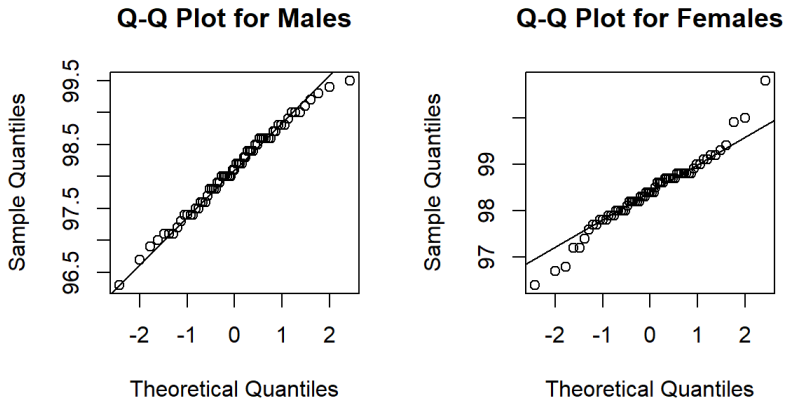
⇨ We create a boxplot for male body temperature data and female body temperature data each.
⇨ Then, we create QQ-Plots for the same.
⇨ Finally, we calculate CI for both male and female data using the t-test.

R-Code:

```
#Section - a
# Drawing boxplots to depict Body Temperatures
boxplot(maleData$body_temperature, femaleData$body_temperature,
  main = "Boxplots of Body Temperatures",
  names = c('Males', 'Females'), ylab = "Temperatures"
)

# Drawing QQPlots for Body Temperature
par(mfrow=c(1,2))
qqnorm(maleData$body_temperature, main = 'Q-Q Plot for Males')
qqline(maleData$body_temperature)
qqnorm(femaleData$body_temperature, main = 'Q-Q Plot for Females')
qqline(femaleData$body_temperature)

# Calculating CI using t-test function for the body temperature values
t.test(maleData$body_temperature, femaleData$body_temperature,
       alternative = 'two.sided', var.equal = F)

```

Output






```
        Welch Two Sample t-test

data:  maleData$body_temperature and femaleData$body_temperature
t = -2.2854, df = 127.51, p-value = 0.02394
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.53964856 -0.03881298
sample estimates:
mean of x mean of y
 98.10462  98.39385
```

Observations

⇨ Looking at the boxplot, we observe that Q1, Median and Q3 are higher for females than of the males so the distribution of females can have a slightly higher mean value than of the males.

- ⇨ Looking at the boxplot, there are more outliers in the females' box plot implies there more variability for them than the males. Hence, we cannot assume equal variances.
- ⇨ As we can see from the Q-Q plots, we can consider the distributions of these body temperature values for both males and females as approximately normal.
- ⇨ Consider,
  - H_0: Difference between means = 0
  - H_1: Difference between means ≠0
- ⇨ The confidence interval we observe because of the function t-test is (-0.53964856, -0.03881298) and the p-value we got is 0.02394.
- ⇨ Since p-value is less than 0.05 and 0 does not lie in the confidence interval, we reject the null hypothesis and hence conclude that the body temperature means of females and males are not equal.
- ⇨ The width of the confidence interval is very small; hence the sample means differ by very small amounts.
- ⇨ And mean of female body temperatures is slightly higher than its counterpart.

b. Do males and females differ in mean heart rate? Answer this question by performing an appropriate analysis of the data, including an exploratory analysis.

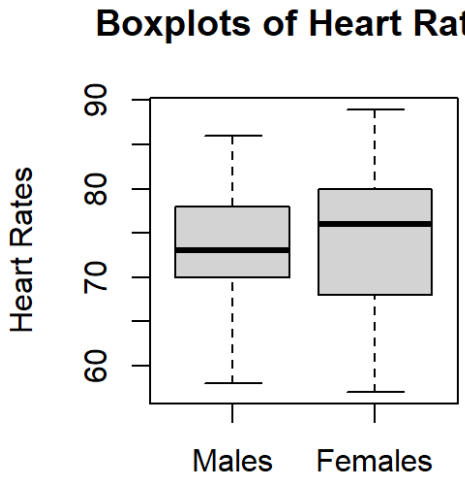
Solution

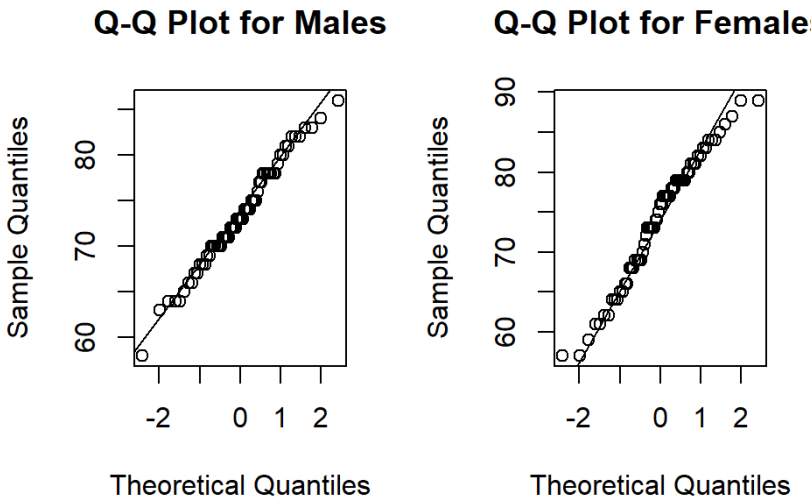
- ⇨ We create a boxplot for male heart rate data and female heart rate data each.
- ⇨ Then, we create QQ-Plots for the same.
- ⇨ Finally, we calculate CI for both male and female data using the t-test.

R-Code:

```
#Section - b
# Drawing boxplots to depict Heart Rates
boxplot(maleData$heart_rate, femaleData$heart_rate,
        main = "Boxplots of Heart Rates",
        names = c('Males', 'Females'), ylab = "Heart Rates"
)

# Drawing QQPlots for Heart Rates
par(mfrow=c(1,2))
qqnorm(maleData$heart_rate, main = 'Q-Q Plot for Males')
qqline(maleData$heart_rate)
qqnorm(femaleData$heart_rate, main = 'Q-Q Plot for Females')
qqline(femaleData$heart_rate)

# Calculating CI using t-test function for the Heart Rates values
t.test(maleData$heart_rate, femaleData$heart_rate,
       alternative = 'two.sided', var.equal = F)

```

Output






```
        Welch Two Sample t-test

data:  maleData$heart_rate and femaleData$heart_rate
t = -0.63191, df = 116.7, p-value = 0.5287
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -3.243732  1.674501
sample estimates:
mean of x mean of y
 73.36923  74.15385
```

Observations

- ⇨ Looking at the boxplot, Q1 for females is less than Q1 for males, but this is not the case for median and Q3 as those values are higher for females than for the males. The values in females seem more stretched out so variability seems to be more.
- ⇨ As we can see from the Q-Q plots, we can consider the distributions of these heart rate values for both males and females as approximately normal.
- ⇨ Consider,
  - H_0: Difference between means = 0
  - H_1: Difference between means ≠0
- ⇨ The confidence interval we observe as a result of the function t-test in R is (-3.243732, 1.674501) and the p-value we got is 0.5287.
- ⇨ Since p-value is greater than 0.05 and the value 0 lies in the confidence interval, we accept the null hypothesis and hence conclude that the heart rate value means of females and males are equal.

c. Is there a linear relationship between body temperature and heart rate? Does this relationship depend on gender? Answer these questions by performing an appropriate analysis of the data, including an exploratory analysis.

Solution

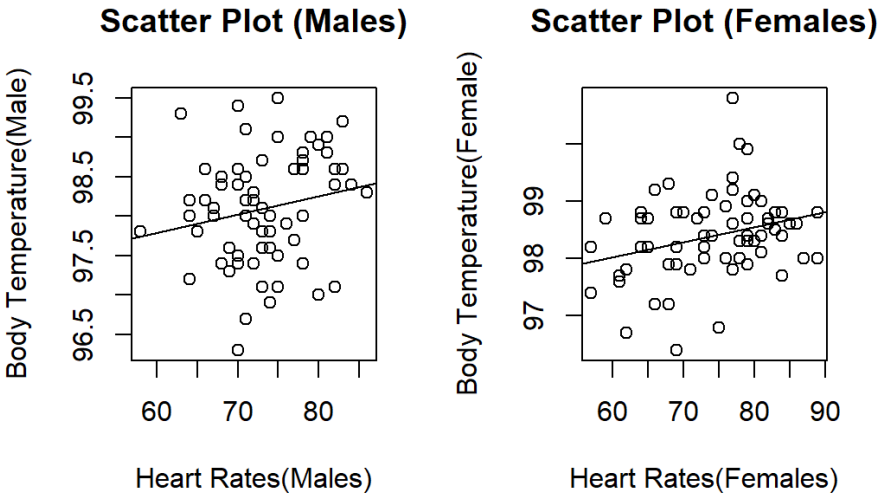
- ⇨ We can draw a scatterplot and draw a regression line that reflects the relationship between body temperature and heart rates for males and females.

R-Code:

```
#Section - c
# Finding the correlation values between body temperatures and heart rates
cor(maleData$body_temperature, maleData$heart_rate)
cor(femaleData$body_temperature, femaleData$heart_rate)

#drawing the scatter plots for the body temperature and heart rate values for males and females
par(mfrow=c(1,2))
plot(maleData$heart_rate, maleData$body_temperature, pch=1,
     main='Scatter Plot (Males)',
     xlab = "Heart Rates(Males)", ylab="Body Temperature(Male)")
abline(lm(maleData$body_temperature~maleData$heart_rate))
plot(femaleData$heart_rate, femaleData$body_temperature, pch=1,
     main='Scatter Plot (Females)',
     xlab = "Heart Rates(Females)", ylab="Body Temperature(Female)")
abline(lm(femaleData$body_temperature~femaleData$heart_rate))
```

Output

```
> cor(maleData$body_temperature, maleData$heart_rate)
[1] 0.1955894
> cor(femaleData$body_temperature, femaleData$heart_rate)
[1] 0.2869312
```



Observations

- ⇨ As we can see from the graph, the line drawn has a slope which is greater than 0. This suggests positive association of correlation between body temperature and heart rate values.
- ⇨ Based on the graph, we can assume that the strength of the linear relationship is weak.
- ⇨ Correlation between body temperature and heart rate for males is: 0.1955894
- ⇨ Correlation between body temperature and heart rate for females is: 0.2869312
- ⇨ As we know that the larger the value the stronger the correlation, Hence we conclude here that the relationship between the body temperature and heart rates is weak.
- ⇨ Since the correlation value for females is higher than males, we can say that that for females the correlation between body temperature and heart rate is a bit stronger than for the males.

Question 2:

The goal of this exercise to see how large n should be for the large-sample and the (parametric) bootstrap percentile method confidence intervals for the mean of an exponential population to be accurate. To be specific, let X1, . . . , Xn represent a random sample from an exponential (λ) distribution. Note that this distribution is skewed and its mean is μ = 1/λ. We can construct two confidence intervals for μ — one the large-sample z-interval (interval 1) and the other a (parametric) bootstrap percentile method interval (interval 2). We would like to investigate their accuracy, i.e., how close their estimated coverage probabilities are to the assumed nominal level of confidence, for various combinations of (n, λ). This investigation will focus on 1 − α = 0.95, λ = 0.01, 0.1, 1, 10 and n = 5, 10, 30, 100. Thus, we have a total of 4 ∗ 4 = 16 combinations of (n, λ) to investigate.

a. **For a given setting, compute Monte Carlo estimates of coverage probabilities of the two intervals by simulating appropriate data, using them to construct the two confidence intervals, and repeating the process 5000 times.**

Solution:

- ⇨ To simulate Monte Carlo estimates and construct confidence intervals, we will first create some functions:
  - checkzci - Takes n and λ vales as input parameters, simulates a sample, constructs an interval, and returns whether the true mean exists within the confidence interval.
  - zproportion - takes n and λ vales as input parameters, calls the checkzci function 5000 times and calculates the coverage probabilities.
  - mean.star – samples from a distribution and returns the mean.
  - checkbci – using the n and λ given as input parameters, it calls the mean.star function 1000 times and forms the confidence interval and returns whether the true mean is present in the interval.
  - bproportion - takes n and λ as input parameters, constructs a parametric initial bootstrap sample and calls checkbci 5000 times and calculates the coverage probabilities.
- ⇨ Then, we will calculate the Z-interval and bootstrap interval values for n = 5 and lambda = 0.01

R-Code

```r
checkzci <- function(n, lambda) {
  U <- rexp(n,lambda)
  lb <- mean(U) - qnorm(0.975) * sd(U) / sqrt(n)
  ub <- mean(U) + qnorm(0.975) * sd(U) / sqrt(n)
  tm = 1/lambda
  if(ub > tm & lb < tm) {
    return (1)
  }
  else {
    return (0)
  }
}

zproportion <- function(n, lambda) {
  values <- replicate(5000, checkzci(n, lambda))
  ones <- values[which (values == 1)]
  return (length(ones)/5000)
}
```

```r
mean.star <- function(n,lambda) {
  u.star <- rexp(n, lambda)
  return (mean(u.star))
}

checkbci <- function(n, lambda) {
  U <- rexp(n,lambda)
  tm <- 1/lambda
  lambda1 = 1/mean(U)
  V <- replicate(1000, mean.star(n,lambda1))
  bound <- sort(V)[c(25, 975)]
  if(bound[2] > tm & bound[1] < tm) {
    return (1)
  }
  else {
    return (0)
  }
}
```

```r
bproportion <- function(n, lambda) {
  values <- replicate(5000, checkbci(n, lambda))
  ones <- values[which (values == 1)]
  return (length(ones)/5000)
}

# Calculating z-interval and bootstrap interval for n = 5 and lambda = 0.01
zproportion(5, 0.01)
bproportion(5, 0.01)
```

Output

```
> zproportion(5, 0.01)
[1] 0.8034
> bproportion(5, 0.01)
[1] 0.8978
```

b. Repeat (a) for the remaining combinations of (n, λ). Present an appropriate summary of the results.

Solution:

⇨ Repeating the above process for the remaining combinations of n and λ we get the following results.

R-code:

```
# Repeating same process for all n, Lambda values
nVals = c(5, 10, 30, 100)
lambdaVals = c(0.01, 0.1, 1, 10)
zprops = c()
bprops = c()

for(l in lambdaVals) {
  for(n in nVals)  {
    zprops <- c(zprops, zproportion(n, l))
    bprops <- c(bprops, bproportion(n, l))
  }
}

zciMatrix = matrix(zprops, nrow = 4, ncol = 4)
bciMatrix = matrix(bprops, nrow = 4, ncol = 4)
```

```
par(mar=c(2,2,2,2), mfrow = c(2,2))
for(x in 1:4) {
  plot(nVals, zciMatrix[,x], main = paste("L = ", lambdaVals[x]),
       xlab = 'n', ylab ='Proportions', col = 'red', type = 'b',
       xlim = c(1,100), ylim = c(0.8,1)
  )
  lines(nVals, bciMatrix[,x], col = 'blue', type = 'b')
}

par(mar=c(2,2,2,2), mfrow = c(2,2))
for(x in 1:4) {
  plot(lambdaVals, zciMatrix[x,], main = paste("N = ", nVals[x]),
       xlab = 'Lambda', ylab ='Proportions', col = 'red', type = 'b',
       xlim = c(0.01,10), ylim = c(0.8,1)
  )
  lines(lambdaVals, bciMatrix[x,], col = 'blue', type = 'b')
}
```

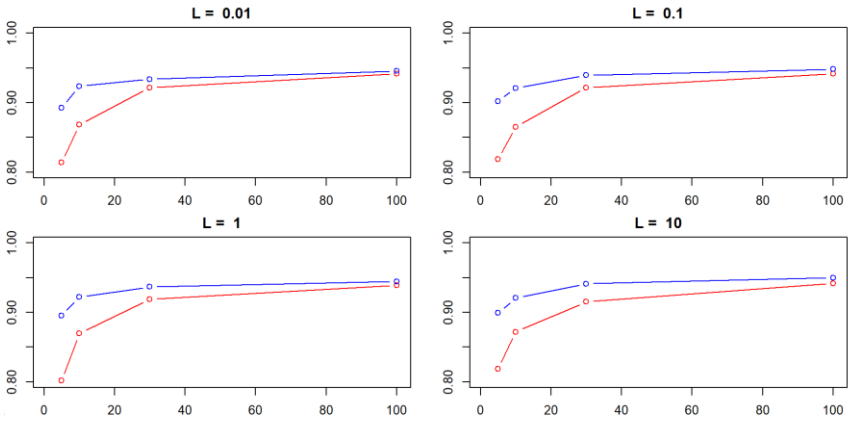
Z-Interval Values:

	L = 0.01	**L = 0.1**	**L = 1**	**L = 10**
N = 5	0.8140	0.8190	0.8018	0.8186
N = 10	0.8684	0.8654	0.8698	0.8720
N = 30	0.9216	0.9218	0.9186	0.9152
N = 100	0.9414	0.9416	0.9390	0.9418

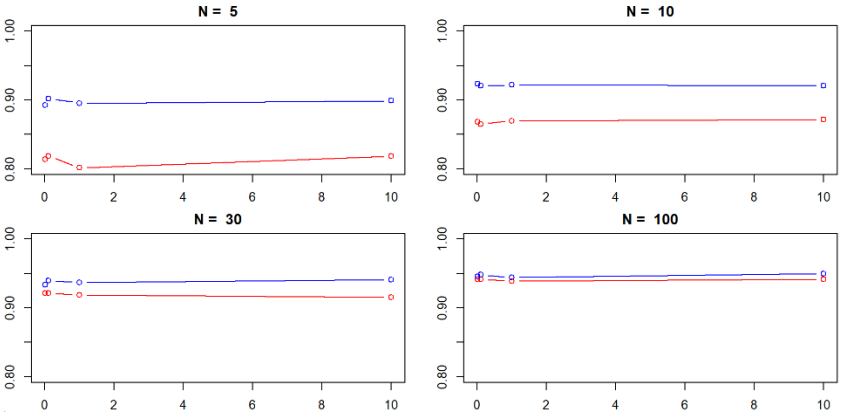
Bootstrap Interval Values:

	L = 0.01	L = 0.1	L = 1	L = 10
N = 5	0.8928	0.9022	0.8954	0.8990
N = 10	0.9236	0.9208	0.9220	0.9206
N = 30	0.9338	0.9396	0.9368	0.9412
N = 100	0.9456	0.9482	0.9444	0.9500

Output:



Graph-1: L is constant and N is variable



Graph-2: N is constant and L is variable

c. Interpret all the results. Be sure to answer the following questions: In case of the large-sample interval, how large n is needed for the interval to be accurate? Likewise, in case of the bootstrap interval, how large n is needed for the interval to be accurate? Do these answers depend on λ? Can we say that one method is more accurate than the other? Which interval would you recommend? Provide justification for all your conclusions.

Solution:

- ⇨ From Graph 1, we see that the graphs don't change drastically when λ is changed, so we can say that the coverage probabilities don't depend on λ.
- ⇨ And we also see that the coverage probabilities we get via bootstrap are higher than those of z-interval method.
- ⇨ From Graph 2, we can conclude that the coverage probabilities depend on n.
- ⇨ Now for the large-sample z-interval, we see that the coverage probabilities are as accurate, as the coverage probabilities we got from bootstrap method, when n is large (n=100).
- ⇨ For the bootstrap method coverage probabilities, they are on the higher side (approximately) from n=30 onwards.
- ⇨ Considering all the graphs, we can say that coverage probabilities we got from bootstrap method are higher for every combination of (n, λ) than for the large-sample z-interval method, hence bootstrap method is more accurate even for the low values of n.
- ⇨ Hence the bootstrap method is recommended.

d. Do your conclusions in (c) depend on the specific values of λ that were fixed in advance? Explain.

Solution:

- ⇨ We can observe that bootstrap samples offer a better probability coverage compared to the Z interval.
- ⇨ Our conclusion depends on primarily the values of N compared to the value of 0.1 for the Lambda.

```
> zciMatrix
       [,1]   [,2]   [,3]   [,4]
[1,] 0.8140 0.8190 0.8018 0.8186
[2,] 0.8684 0.8654 0.8698 0.8720
[3,] 0.9216 0.9218 0.9186 0.9152
[4,] 0.9414 0.9416 0.9390 0.9418
```

```
> bciMatrix
       [,1]   [,2]   [,3]   [,4]
[1,] 0.8928 0.9022 0.8954 0.8990
[2,] 0.9236 0.9208 0.9220 0.9206
[3,] 0.9338 0.9396 0.9368 0.9412
[4,] 0.9456 0.9482 0.9444 0.9500
```

- ⇨ We can see that the bootstrap proportions are greater for almost all combinations for N less than 30 and for lambda values other than 0.1.