Mini Project 04
Mini Project Group 31
Group Members
Dhairya Pinakin Desai: DPD140130

Question 1(a):

In order to determine the difference between the means of the body temperatures of male and female individuals, we do exploratory data analysis (EDA) on our data using the formal statistical approach of the Z-test.

```
> # Setting working directory to Project folder.
> setwd("C:/Users/dpd140130/OneDrive - The University of Texas at Dallas/CS 6313/Projects/05")
> getwd()
```

We are using z.test() function from the BSDA library which is used to perform a one-sample (or two-sample) z-test to calculate mean.

```
# install the 'BSDA' package first to load it.
> install.packages("BSDA")
> library(BSDA)
```

Creating two vectors to store Male and Female data.

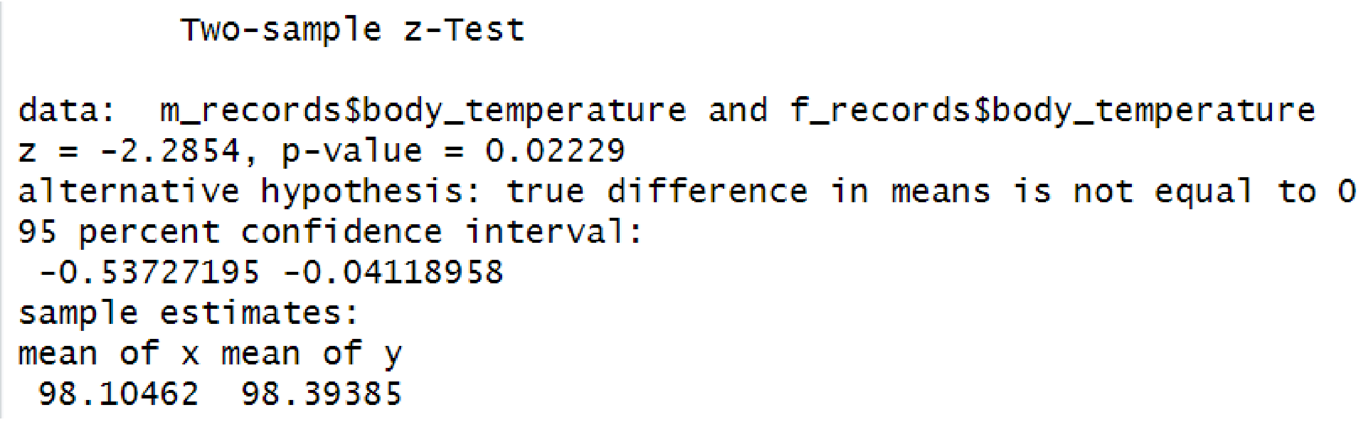
```
# Observing our .csv file
> records <- read.csv("bodytemp-heartrate.csv")
> m_records <- records[records$gender==1,]
> f_records <- records[records$gender==2,]
```

Our sample set has 65 observations, which is significantly more than 30 (S>>30), thus according to the Law of Large Numbers we may infer that the sample follows Normal Distribution.
Therefore, we specify the following parameters while running the hypothesis testing using z.test():

x,y: Male and Female subjects' data.
alternative: The alternative hypothesis for the test. It can be 'greater', 'less', or 'two. sided' based on the alternative hypothesis.
Null Hypothesis H_0: μM = μF

mu: 0.
sigma.x: It represents the population standard deviation for the x sample.
sigma.y: It represents the population standard deviation for the y sample.
conf. level: confidence level of the interval , default : 95%.

```
>z.test(x=m_records$body_temperature,y=f_records$body_temperature,mu=0,sigma.x=sd(m_records$body_temperature),sigma.y=sd(f_records$body_temperature),alternative="two.sided")
```



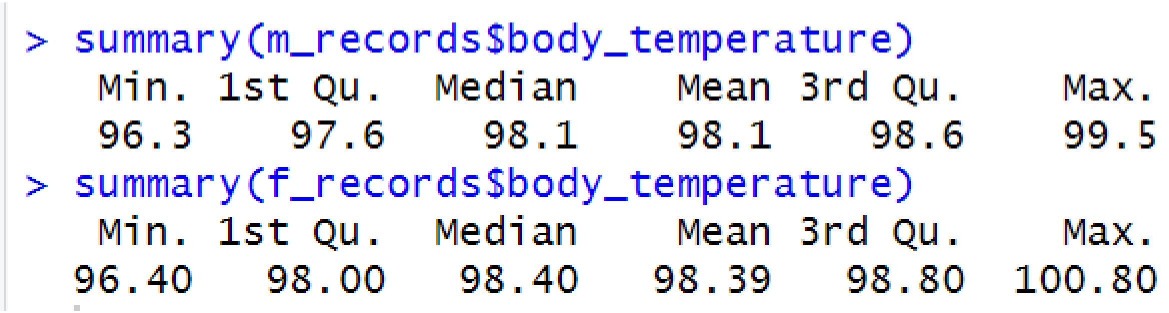
```
        Two-sample z-Test

data:  m_records$body_temperature and f_records$body_temperature
z = -2.2854, p-value = 0.02229
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.53727195 -0.04118958
sample estimates:
mean of x mean of y 
 98.10462  98.39385 
```

Conclusion:
According to p-value approach (alpha: level of confidence = 0.05), our p-value (0.02229) from z-test is lesser than the alpha = 0.05. So, the null hypothesis, i.e. the difference in mean of body temperature of Male and Female subjects is equal, is rejected. Therefore, **Alternate Hypothesis is accepted.**

Our conclusion is also proved by the summary statistics.

```
> summary(m_records$body_temperature)
> summary(f_records$body_temperature)
```

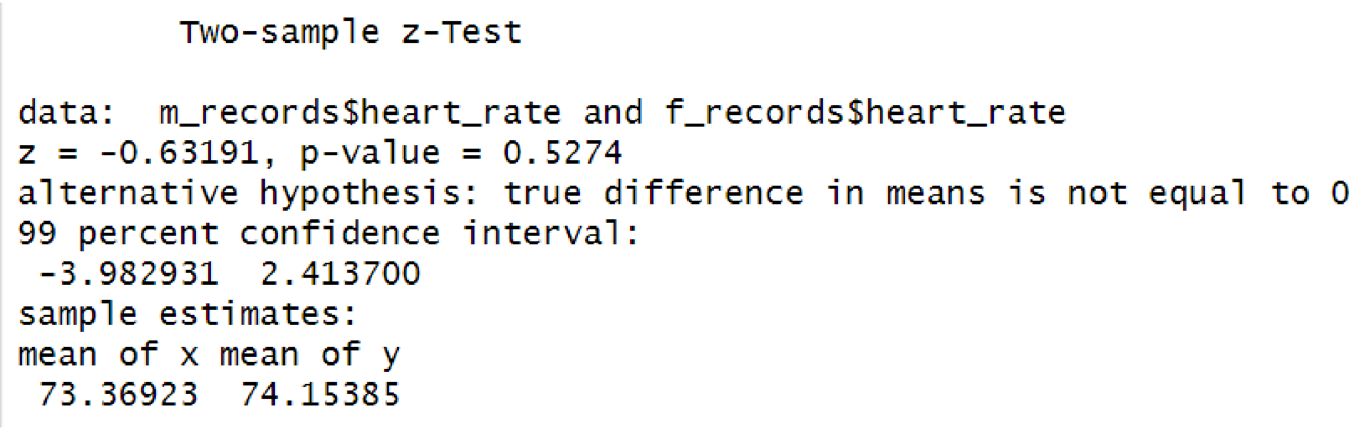


```
> summary(m_records$body_temperature)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   96.3    97.6    98.1    98.1    98.6    99.5 
> summary(f_records$body_temperature)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  96.40   98.00   98.40   98.39   98.80  100.80 
```

Question 1(b):

The procedure will be the same for analysing the mean heart rate measure for the male and female individuals. If both datasets adhere to a Normal Distribution according to the Law of Large Numbers, we will use the z-test to assess whether or not there is a difference in the mean HR of the two datasets.

```
>z.test(x=m_records$heart_rate,y=f_records$heart_rate,mu=0,sig
ma.x=sd(m_records$heart_rate),sigma.y=sd(f_records$heart_rate)
,alternative="two.sided",conf.level = 0.99)
```



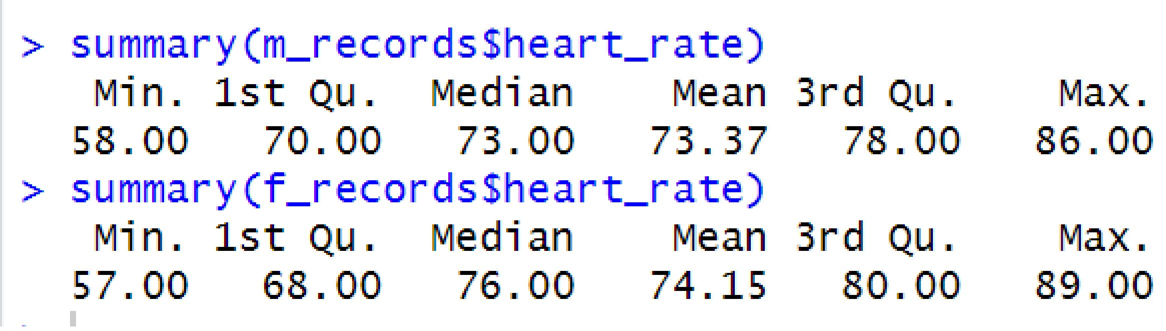
```
        Two-sample z-Test

data:  m_records$heart_rate and f_records$heart_rate
z = -0.63191, p-value = 0.5274
alternative hypothesis: true difference in means is not equal to 0
99 percent confidence interval:
 -3.982931  2.413700
sample estimates:
mean of x mean of y
 73.36923  74.15385
```

Conclusion:
According to p-value approach (alpha: level of confidence = 0.01), our p-value (0.5274) from z-test is lesser than the alpha = 0.01. So, the null hypothesis, i.e. the difference in mean of body temperature of Male and Female subjects is equal, is accepted. Therefore, **Null Hypothesis is accepted.**

Our conclusion is also proved by the summary statistics.

```
> summary(m_records$heart_rate)
> summary(f_records$heart_rate)
```



```
> summary(m_records$heart_rate)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  58.00   70.00   73.00   73.37   78.00   86.00 
> summary(f_records$heart_rate)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  57.00   68.00   76.00   74.15   80.00   89.00 
```

Question 1(c):

For both genders, we will create a scatter plot with confidence interval band to show the relationship between body temperature and heart rate.

For Male Participants:

Pearson's Correlation Coefficient:

The strength and direction of the linear relationship between two variables are measured using this method. It can take values between -1 and 1, and is represented by the symbol " ρ".

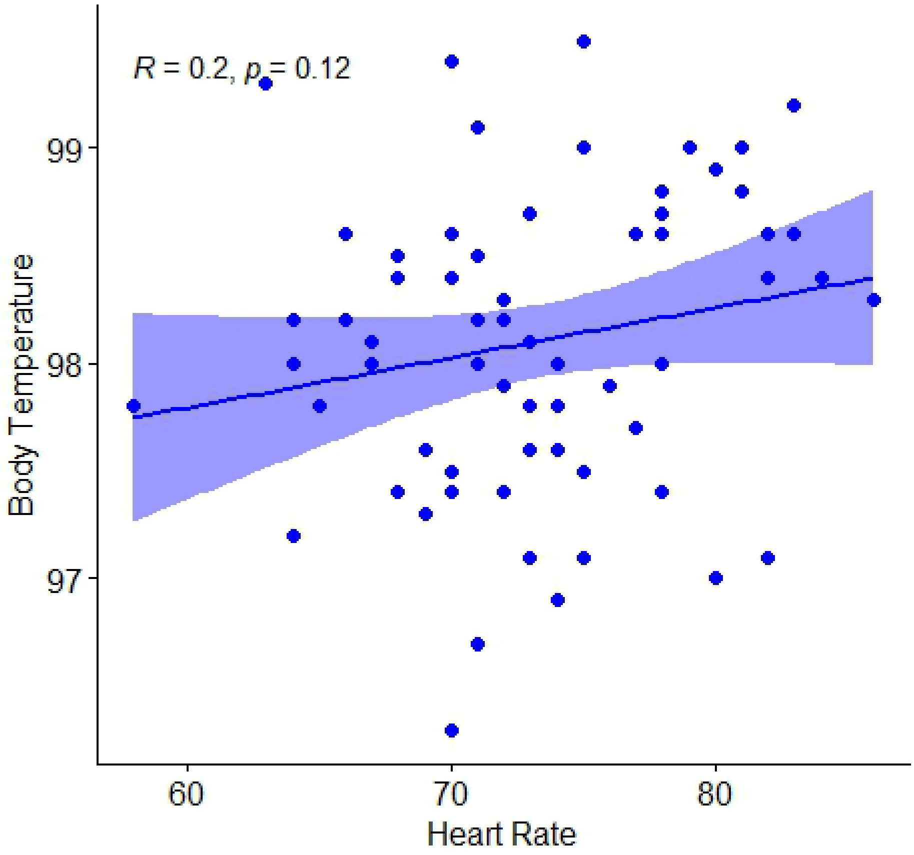
If the value is 1, there is a perfect positive correlation, which means that as one variable rises, the other rises linearly as well. If the value is -1, there is a perfect negative correlation, which means that as one measure rises, the other one falls linearly. No correlation, or a value of 0, denotes the absence of a linear relationship between the variables.

To determine the correlation between the heart rate and body temperature of Male Participants, we use cor() function.

```
> rho <- cor(m_records$heart_rate,m_records$body_temperature)
# finding Pearson's correlation coeff.
> rho
[1] 0.1955894
```

According to our analysis of the value of "rho," which is close to 0.2, there is only a weak positive correlation between heart rate and body temperature of Male Participants. No significant effect on correlation was observed with change in gender.

```
> # found the point estimate of rho.
> ggpubr::ggscatter(m_records, x = "heart_rate", y = "body_temperature", add = "reg.line", conf.int = TRUE, cor.coef = TRUE, cor.method = "pearson", xlab = "Heart Rate", ylab = "Body Temperature", col = "blue")
```



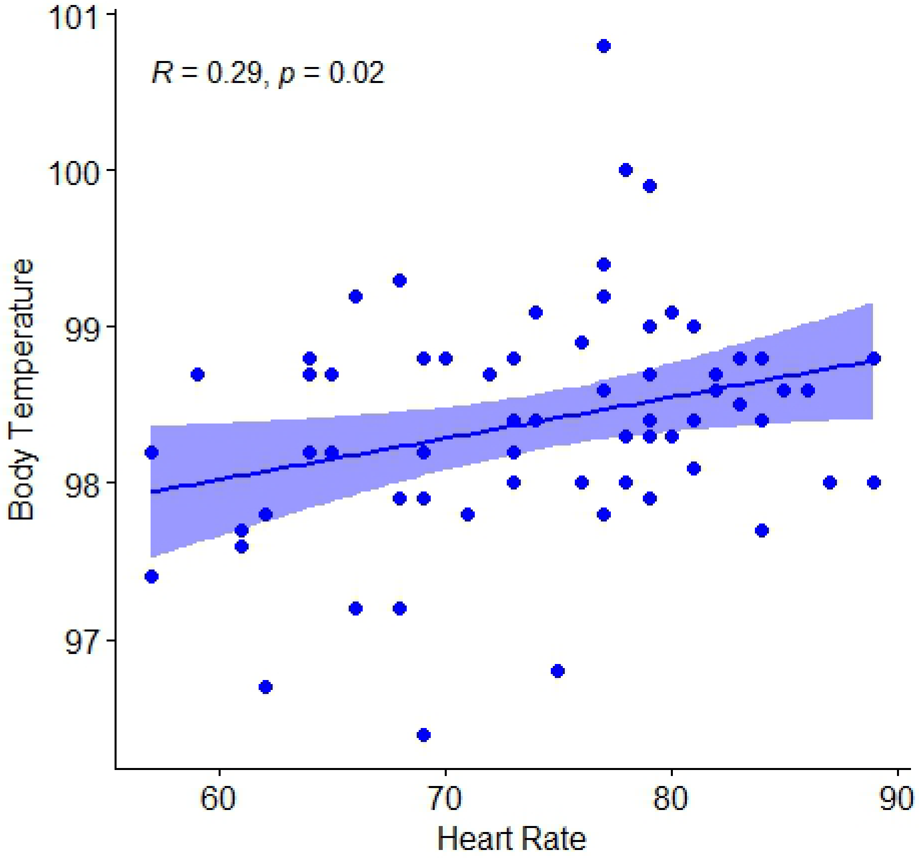
<u>For Female Participants:</u>

To determine the correlation between the heart rate and body temperature of Female Participants, we use cor() function.

```
> rho <- cor(f_records$heart_rate,f_records$body_temperature)
# finding Pearson's correlation coeff.
> rho
[1] 0.2869312
```

According to our analysis of the value of "rho," which is close to 0.2, there is only a weak positive correlation between heart rate and body temperature of Female Participants. No significant effect on correlation was observed with change in gender.

```
> # found the point estimate of rho.
> ggpubr::ggscatter(f_records, x = "heart_rate", y = "body_temperature", add = "reg.line", conf.int = TRUE, cor.coef = TRUE, cor.method = "pearson", xlab = "Heart Rate", ylab = "Body Temperature", col = "blue")
```



R Code:

```r
# STATS-Mini-Project-5
# Setting working directory to Proj-5 folder.
setwd("C:/Users/dpd140130/OneDrive - The University of Texas at
Dallas/CS 6313/Projects/05")
getwd()
#install.packages("BSDA")
library(BSDA)

# Observing our .csv file
records <- read.csv("bodytemp-heartrate.csv")
m_records <- records[records$gender==1,]
f_records <- records[records$gender==2,]

# Question-1(a)
# Hypothesis Testing for body_temp metric for M and F data.
z.test(x=m_records$body_temperature,y=f_records$body_temperature,mu=0,
sigma.x=sd(m_records$body_temperature),sigma.y=sd(f_records$body_tempe
rature),alternative="two.sided")
summary(m_records$body_temperature)
summary(f_records$body_temperature)

# Question-1(b)
# Hypothesis Testing for heart_rate metric for M and F data.
z.test(x=m_records$heart_rate,y=f_records$heart_rate,mu=0,sigma.x=sd(m
_records$heart_rate),sigma.y=sd(f_records$heart_rate),alternative="two
.sided",conf.level = 0.99)
summary(m_records$heart_rate)
summary(f_records$heart_rate)

# Question-1(c)
#For Male Participants:
rho <- cor(m_records$heart_rate,m_records$body_temperature)
# finding Pearson's correlation coeff.
rho
# found the point estimate of rho.
ggpubr::ggscatter(m_records, x = "heart_rate", y = "body_temperature",
add = "reg.line", conf.int = TRUE, cor.coef = TRUE, cor.method =
"pearson", xlab = "Heart Rate", ylab = "Body Temperature", col =
"blue")

#For Female Participants:
rho <- cor(f_records$heart_rate,f_records$body_temperature)
# finding Pearson's correlation coeff.
rho
# found the point estimate of rho.
ggpubr::ggscatter(f_records, x = "heart_rate", y = "body_temperature",
add = "reg.line", conf.int = TRUE, cor.coef = TRUE, cor.method =
"pearson", xlab = "Heart Rate", ylab = "Body Temperature", col =
"blue")
```

Question 2(a):

Computing 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.

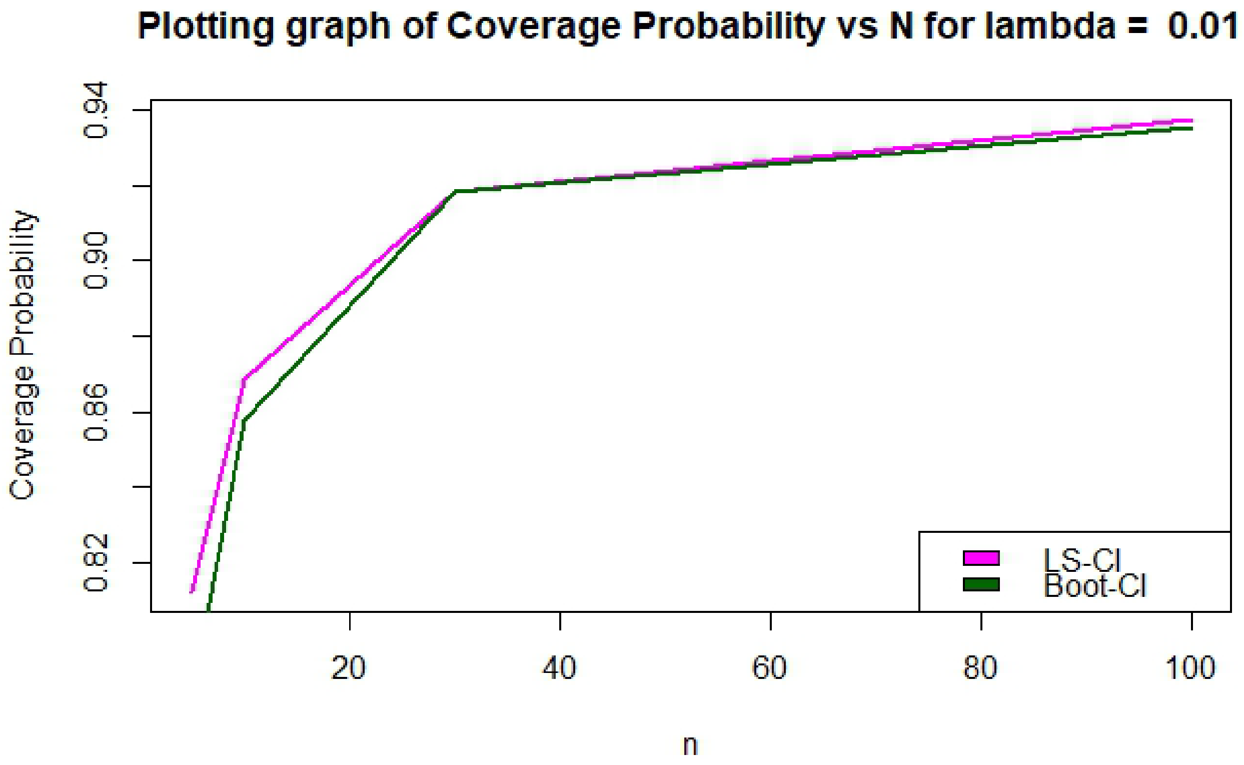
For n = 30 and lambda = 0.1, repeating the process 5000 times.

```
> n <- 30
> lamda <- 0.1
> for (i in 1:5000)
{
# Generating data which follows Exponential Distribution.
data <- rexp(n, lamda)
# Calculating the Confidence Interval using Large Sample Mean
formula.
lwr.z <- mean(data) - (qnorm(.975)*sd(data)/sqrt(n))
upr.z <- mean(data) + (qnorm(.975)*sd(data)/sqrt(n))
# Calculating the Confidence Interval using Parametric
Bootstrap Percentile.
BootR <- boot(data=data, statistic = mean.boot, R= 999)
lwr.boot <- quantile(BootR$t,.025)
upr.boot <- quantile(BootR$t,.975)
# For each of the 5000 iterations, creating vectors to store
the CI generated by large sample z-interval and parametric
bootstrap method.
z_coverage[i] <- ((1/lamda) >= lwr.z) & ((1/lamda) <= upr.z)
boot_coverage[i] <- ((1/lamda) >= lwr.boot) & ((1/lamda) <=
upr.boot)
}
```

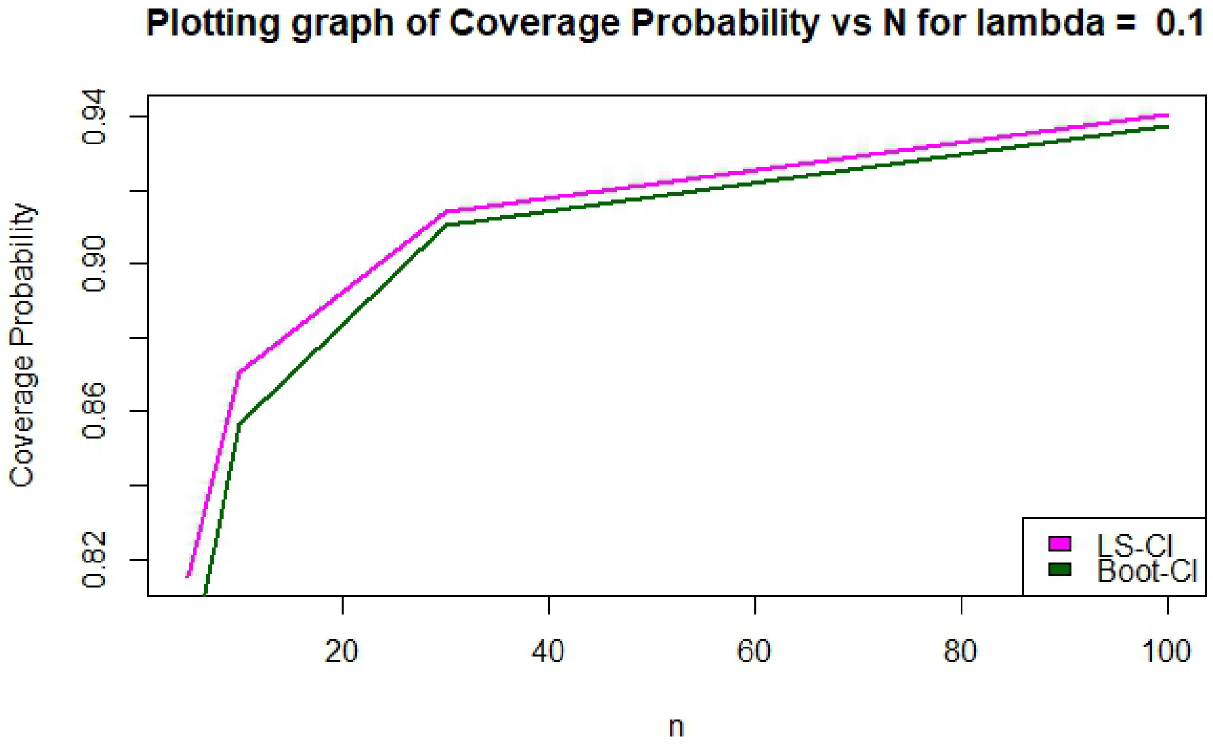
Question 2(b):

Now repeating the above process for all the combinations of (n, lamda).
By keeping lambda constant, we iterate through all the 4 values of n, computing 5000 C.I. for each combination.
Finally, we plot these in the form of a line plot for ease of visualisation.

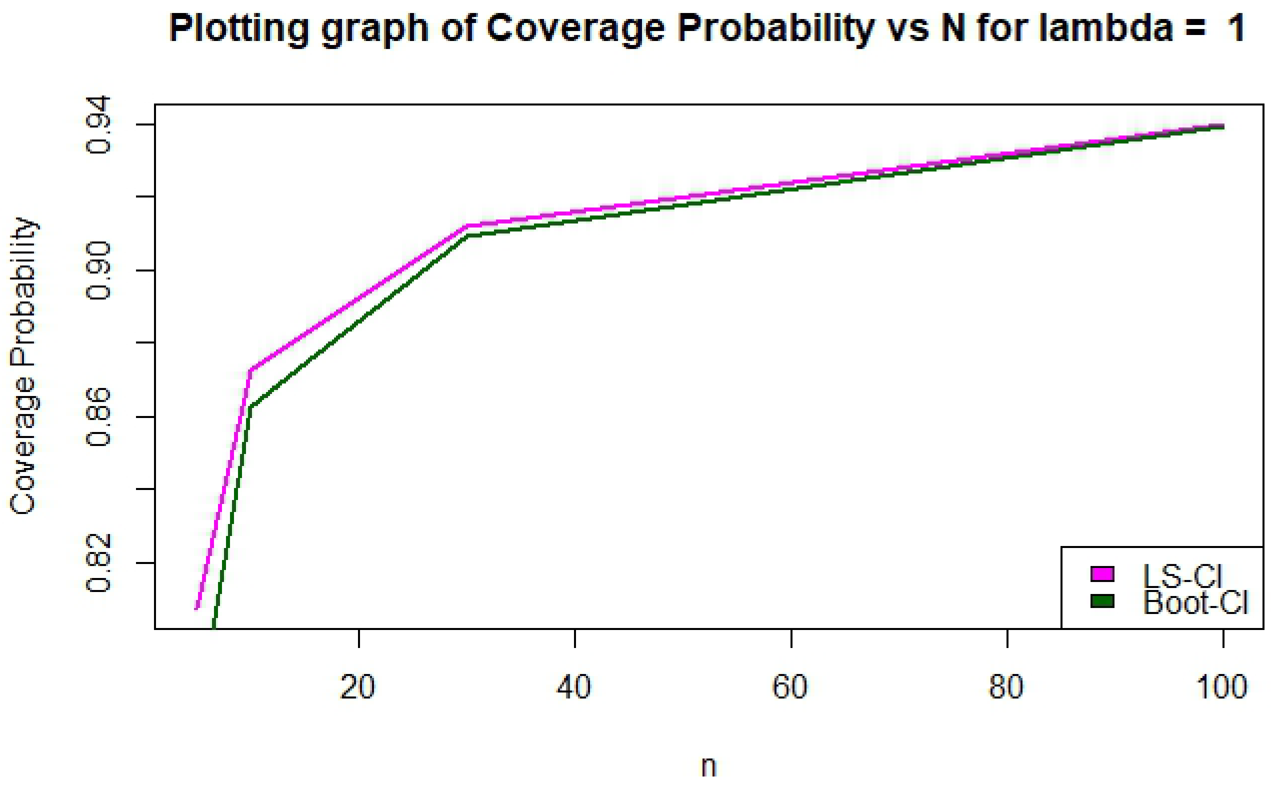
R code is given at the end of this question.

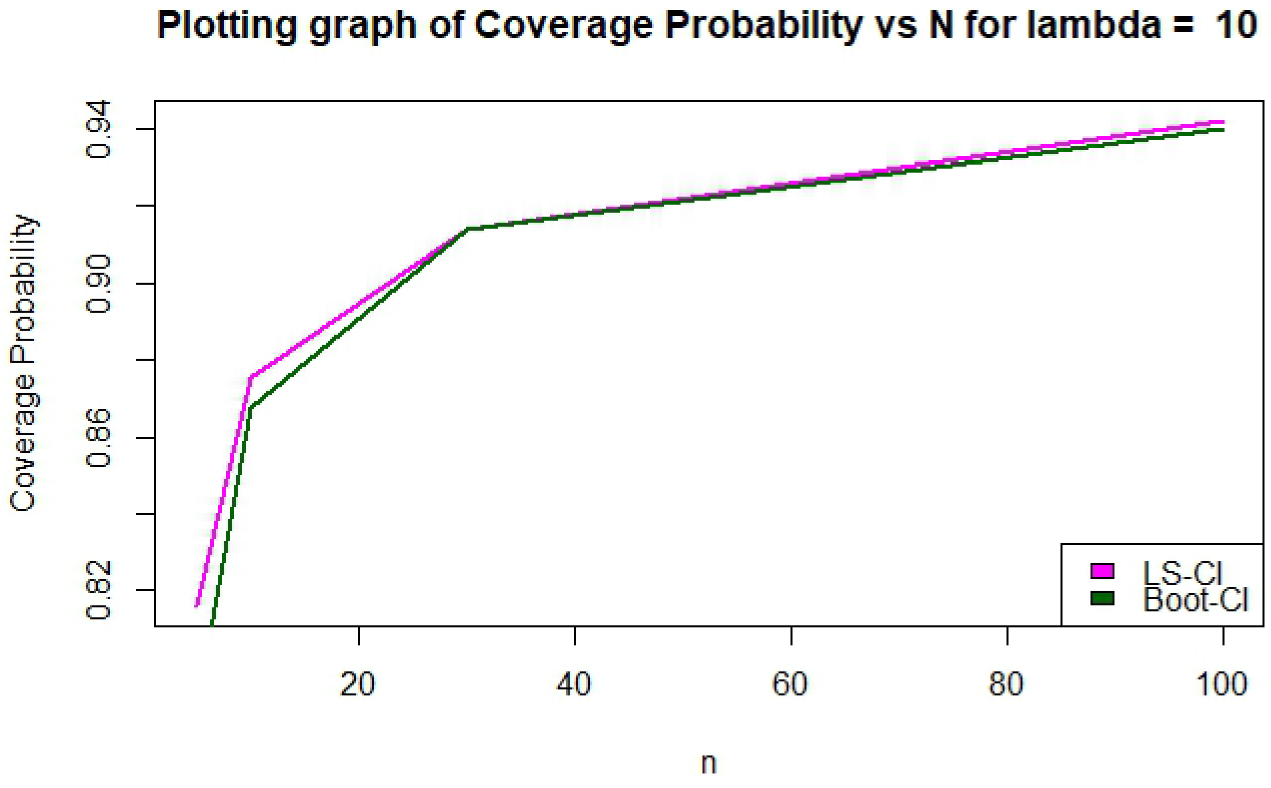


Plotting graph of Coverage Probability vs N for lambda = 0.01

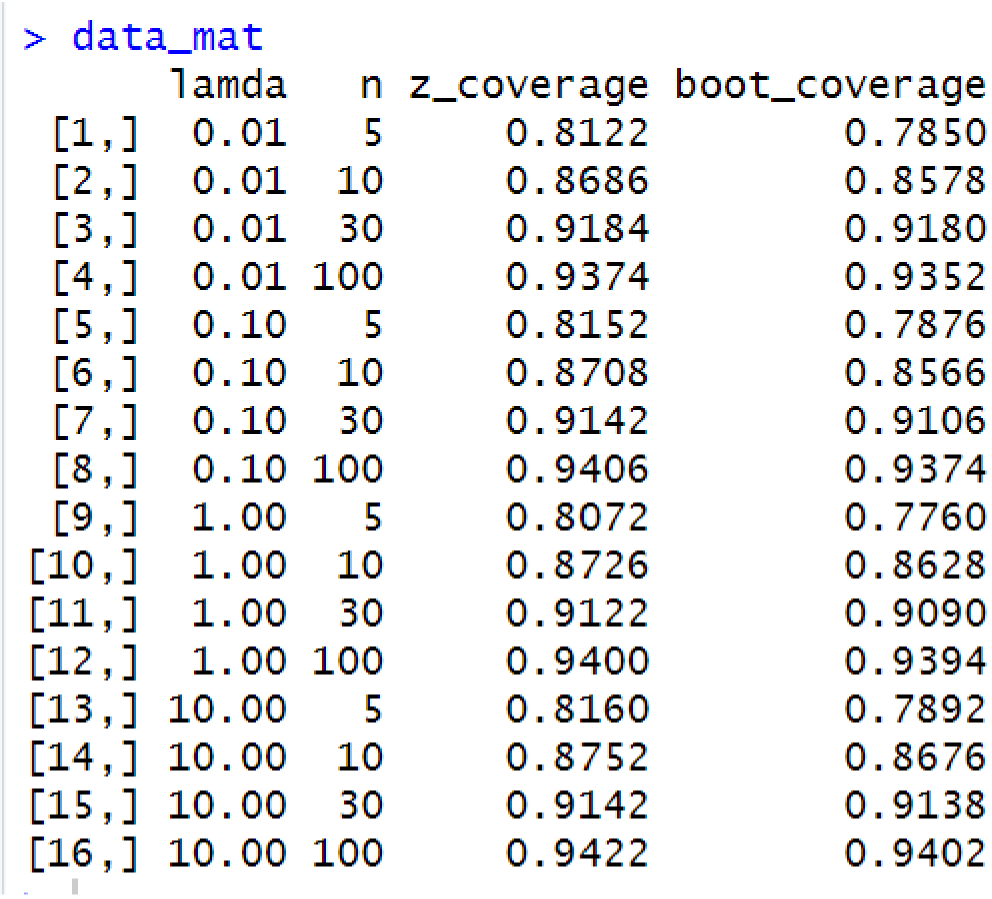


Plotting graph of Coverage Probability vs N for lambda = 0.1


Plotting graph of Coverage Probability vs N for lambda = 1
Coverage Probability
0.82
0.86
0.90
0.94
20
40
60
80
100
n
LS-CI
Boot-CI


Plotting graph of Coverage Probability vs N for lambda = 10
Coverage Probability
0.82
0.86
0.90
0.94
20
40
60
80
100
n
LS-CI
Boot-CI

Now displaying the values in the form of a matrix, containing all 16 combinations of (n, lamda):



```
> data_mat
      lamda   n z_coverage boot_coverage
 [1,]  0.01   5     0.8122        0.7850
 [2,]  0.01  10     0.8686        0.8578
 [3,]  0.01  30     0.9184        0.9180
 [4,]  0.01 100     0.9374        0.9352
 [5,]  0.10   5     0.8152        0.7876
 [6,]  0.10  10     0.8708        0.8566
 [7,]  0.10  30     0.9142        0.9106
 [8,]  0.10 100     0.9406        0.9374
 [9,]  1.00   5     0.8072        0.7760
[10,]  1.00  10     0.8726        0.8628
[11,]  1.00  30     0.9122        0.9090
[12,]  1.00 100     0.9400        0.9394
[13,] 10.00   5     0.8160        0.7892
[14,] 10.00  10     0.8752        0.8676
[15,] 10.00  30     0.9142        0.9138
[16,] 10.00 100     0.9422        0.9402
```

R Code:

```
library(boot)
n <- c(5,10,30,100)
lamda <- c(0.01,0.1,1,10)

lwr.z = numeric(0)
upr.z = numeric(0)
lwr.boot = numeric(0)
upr.boot= numeric(0)
z_coverage = numeric(0)
boot_coverage = numeric(0)

mean.boot<-function(rec,sub){
  return(mean(rec[sub]))
  }

data_mat = matrix(,nrow = 1,ncol = 4,dimnames = list(c(1),
c("lamda","n","z_coverage","boot_coverage")))
data_mat = data_mat[-c(1),]

for(param in lamda){
  CI_LLN = numeric(0)
  CI_BootPerCI = numeric(0)
  k=1
  for(datasize in n){
    for (i in 1:5000){

      # Generating data which follows Exponential Distribution.
      data <- rexp(datasize, param)

      # Calculating the Confidence Interval using Large Sample Mean formula.
      lwr.z <- mean(data) - (qnorm(.975)*sd(data)/sqrt(datasize))
      upr.z <- mean(data) + (qnorm(.975)*sd(data)/sqrt(datasize))

      # Calculating the Confidence Interval using Parametric Bootstrap Percentile.
      BootR <- boot(data=data, statistic = mean.boot, R= 999)
      lwr.boot <- quantile(BootR$t,.025)
      upr.boot <- quantile(BootR$t,.975)
      # For each of the 5000 iterations, creating vectors to store the CI generated
      by large sample z-interval and parametric bootstrap method.
      z_coverage[i] <- ((1/param) >= lwr.z) & ((1/param) <= upr.z)
      boot_coverage[i] <- ((1/param) >= lwr.boot) & ((1/param) <= upr.boot)
    }

    CI_LLN[k] <- mean(z_coverage)
    CI_BootPerCI[k] <- mean(boot_coverage)

    data_mat = rbind(data_mat,c(param,datasize,CI_LLN[k],CI_BootPerCI[k]))
    k=k+1
  }
  plot(n,CI_LLN,type = 'l', col="magenta",lwd=2, ylab="")
  title(main = paste("Plotting graph of Coverage Probability vs N for lambda =
",param), ylab = 'Coverage Probability')
  lines(n,CI_BootPerCI,type = 'l', col="darkgreen",lwd=2)
  legend("bottomright", legend = c('LS-CI','Boot-CI'), fill =
c('magenta','darkgreen'))
}

data_mat
```

Question 2(c):

Q1) In case of the large-sample interval, how large n is needed for the interval to be accurate?

- As we can see from our plots, the sample size must be bigger than 30 (n>30) in order for the Confidence Intervals produced by the Large Sample z-test to be accurate.

Q2) Likewise, in case of the bootstrap interval, how large n is needed for the interval to be accurate?

- We are aware that the Bootstrapping approach is best used when the dataset is sufficiently large, n > 30, because only then may numerous samples be created from the original one using resampling techniques.

Q3) Do these answers depend on λ? Can we say that one method is more accurate than the other?

- We can observe from the created n, lambda matrix that the solutions don't appear to be dependent on lambda's value. We cannot make a judgment on whether approach is more accurate based on the data. However, if a big sample is provided, the z-test will often show to be more accurate.

Q4) Which interval would you recommend?

- Since both are beneficial in their own circumstances, there is no suggestion to be given. By examining the dataset that has been given to him, the user must make that determination.

Question 2(d):

No, as shown by the data in the table, the results drawn in the inquiry are not reliant on the value of the lambda parameter provided to us.