Mini Project #5

Group 2.

Group Members:

Manav Gupta [MXG220027]

Shalin Ronakkumar Kaji [SXK220263]

Contribution:

Manav : Solved Question 2.

Shalin : Solved Question 1 and prepared the final draft of the documentation.

Question 1(a):

We conduct Exploratory Data Analysis (EDA) on our data using the Formal Statistical method of Z-Test for finding the difference in means of Male and Female subjects' Body Temperatures.

```
# STATS-Mini-Project-5

# Setting working directory to Proj-5 folder.

> setwd("C:/Users/Shalin Kaji/Desktop/UT-Dallas-Spr23/STATS-DS-Min.Chen/Mini-Proj-5")

> getwd()
```

z.test() function in R from the BSDA library is used to perform a one-sample (or two-sample) z-test for mean.

```
> library(BSDA)
```

Now we create two vectors to split and store the data of the Male and Female subjects separately.

```
# Exploring our .csv file
```

```
> data <- read.csv("bodytemp-heartrate.csv")

> mdata <- data[data$gender==1,]

> fdata <- data[data$gender==2,]
```

Since the number of observations = 65 in our sample set, which being much greater than 30 (S>>30), we can assume that the given sample follows Normal Distribution by the **Law of Large Numbers**.

Hence, on performing the Hypothesis Testing using z.test() we set the following parameters:

x,y: Male and Female subjects' data.

alternative: The alternative hypothesis for the test. It can be 'greater', 'less', 'two. sided' based on the alternative hypothesis.

Null Hypothesis $H_0 : \mu_M = \mu_F$

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%.

```
> print('Mean BodyTemps are unequal as shown by z-test on M-F data.')
> z.test(x=mdata$body_temperature,y=fdata$body_temperature,mu=0,
sigma.x=sd(mdata$body_temperature),sigma.y=sd(fdata$body_temperature),
alternative="two.sided")
```

```
        Two-sample z-Test

data:  mdata$body_temperature and fdata$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
```

Interpreting the results of our z-test:-

Decision Rule: p-value approach (alpha : level of confidence = 0.05)

Since the p-value = 0.02229 is lesser than the alpha = 0.05,

We reject our Null Hypothesis ie. The difference in mean of body temperatures of Male and Female subjects is equal.

The same is also proved by deriving a summary of statistics of both our datasets.

> summary(mdata$body_temperature)

> summary(fdata$body_temperature)

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

Conclusion: Alternate Hypothesis H_A is accepted.

The mean body temperatures of M and F are unequal.

Question 1(b):

For analysing the mean Heart Rate metric of the Male and Female subjects, the process will be the same as above. We will use the **z-test** to determine whether or not there is a difference in mean HR of the two datasets, assuming that both the datasets follow a Normal Distribution by the Law of Large Numbers.

Hypothesis Testing for heart_rate metric for M and F data.

Since number of obv is 65 we can assume that it follows Normal Distribution from the Law of Large Numbers.

```
> print('Mean Heart-Rates are unequal as shown by z-test on M-F data.')
> z.test(x=mdata$heart_rate,y=fdata$heart_rate,mu=0,
sigma.x=sd(mdata$heart_rate),sigma.y=sd(fdata$heart_rate),
alternative="two.sided",conf.level = 0.99)
```

```
        Two-sample z-Test

data:  mdata$heart_rate and fdata$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
```

Interpreting the results of our z-test:-

Decision Rule: p-value approach (alpha : level of confidence = 0.01)

Since the p-value = 0.5274 is greater than the alpha = 0.01,

We accept our Null Hypothesis ie. The difference in mean of heart rates of Male and Female subjects is equal.

However, the opposite is proved by deriving a summary of statistics of our datasets.

```
> summary(mdata$heart_rate)
> summary(fdata$heart_rate)
```

```
> summary(mdata$heart_rate)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  58.00   70.00   73.00   73.37   78.00   86.00
> summary(fdata$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):

To find the relationship between body-temperature and heart-rate we will develop a scatter plot with confidence interval bands for both the genders.

Male subjects:

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

```
> # Question-1(c)
> rho <- cor(mdata$heart_rate,mdata$body_temperature)
> rho
[1] 0.1955894
> # found the point estimate of rho.
> ggpubr::ggscatter(mdata, x = "heart_rate", y = "bod
+                   add = "reg.line", conf.int = TRUE
```

As we can see that the value of *rho* is approximately 0.2, which indicates a near non-existent relationship between the body-temperature and heart-rate for Males. OR it might suggest a **weak positive correlation** if we are provided with a larger sample size. No significant effect on correlation was observed with change in gender.

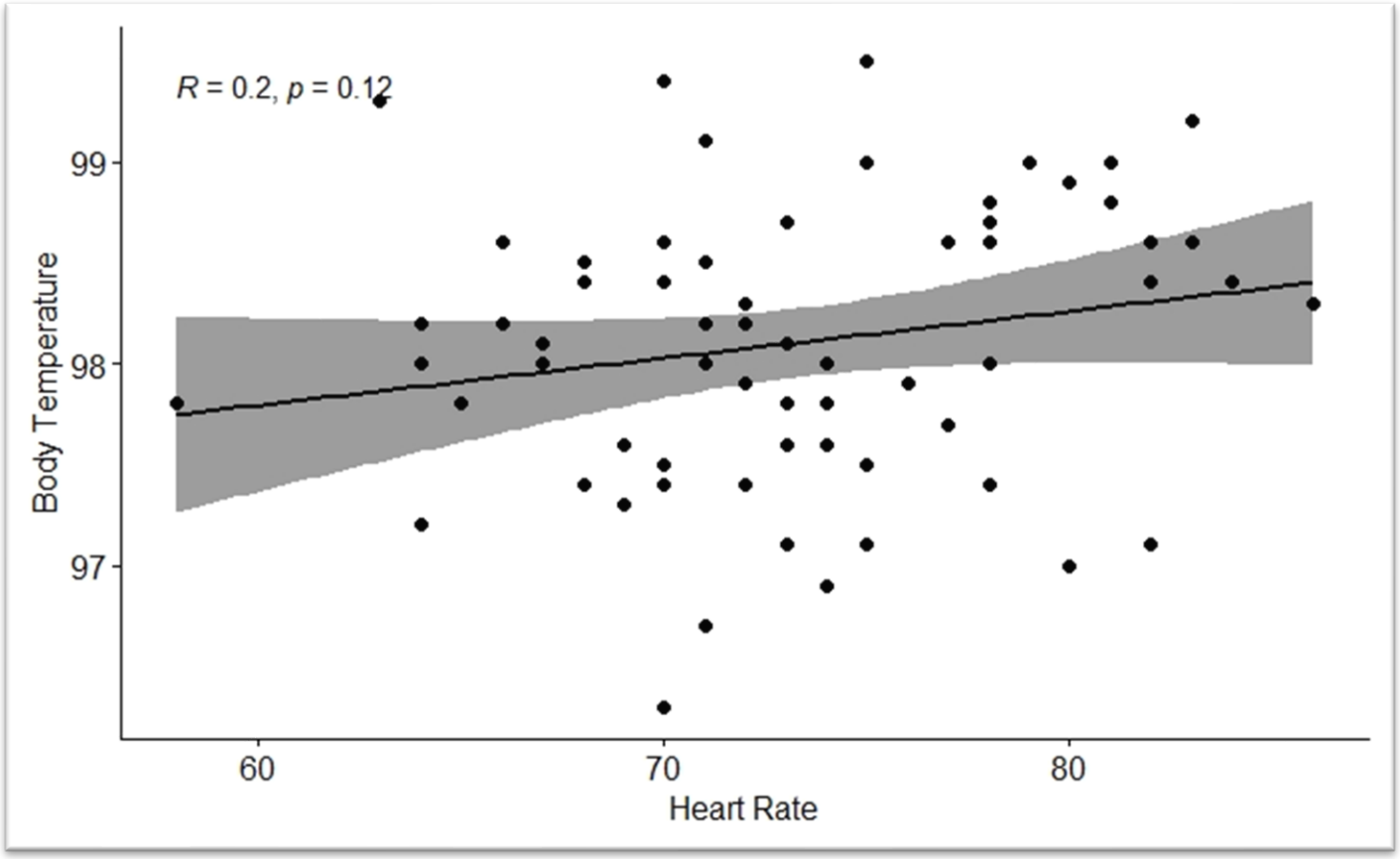


Fig: Scatterplot of Temp vs HR for Male.

Female Subjects:

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

```
> rho <- cor(fdata$heart_rate,fdata$body_temperature) # finding Pearson's correlation coeff.
> rho
[1] 0.2869312
> # found the point estimate of rho.
```

As we can see that the value of *rho* is approximately 0.3, it might suggest a **weak positive correlation** for females.

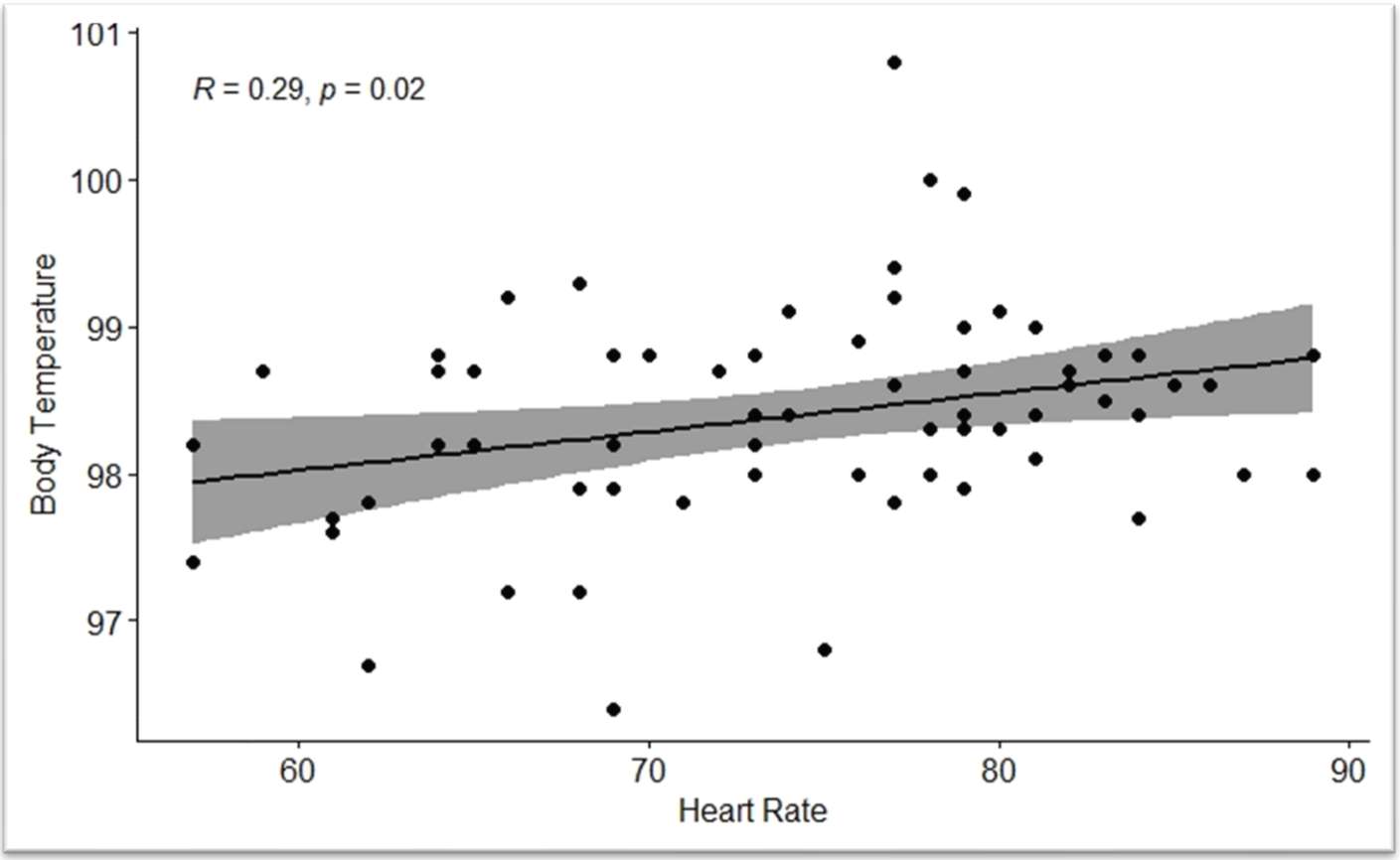


Fig: Scatterplot for Temp vs HR for Females.

R Code:

```
# STATS-Mini-Project-5
# Setting working directory to Proj-5 folder.
setwd("C:/Users/Shalin Kaji/Desktop/UT-Dallas-Spr23/STATS-DS-Min.Chen/Mini-Proj-5")
getwd()
library(BSDA)

# Exploring our .csv file
data <- read.csv("bodytemp-heartrate.csv")
typeof(data$gender)
mdata <- data[data$gender==1,]
fdata <- data[data$gender==2,]
```

R Code:

```
# Question-1(a)

# Hypothesis Testing for body_temp metric for M and F data.

# Since number of obv is 65 we can assume that it follows Normal Distribution
from the Law of Large Numbers.

print('Mean BodyTemps are unequal as shown by z-test on M-F data.')

z.test(x=mdata$body_temperature,y=fdata$body_temperature,mu=0,sigma.x=sd(mdata$bo
dy_temperature),sigma.y=sd(fdata$body_temperature),alternative="two.sided")

summary(mdata$body_temperature)

summary(fdata$body_temperature)

# Question-1(b)

# Hypothesis Testing for heart_rate metric for M and F data.

# Since number of obv is 65 we can assume that it follows Normal Distribution
from the Law of Large Numbers.

print('Mean Heart-Rates are unequal as shown by z-test on M-F data.')

z.test(x=mdata$heart_rate,y=fdata$heart_rate,mu=0,sigma.x=sd(mdata$heart_rate),si
gma.y=sd(fdata$heart_rate),alternative="two.sided",conf.level = 0.99)

summary(mdata$heart_rate)

summary(fdata$heart_rate)

# Question-1(c)

rho <- cor(mdata$heart_rate,mdata$body_temperature) # finding Pearson's
correlation coeff.

rho

# found the point estimate of rho.

ggpubr::ggscatter(mdata, x = "heart_rate", y = "body_temperature",
                  add = "reg.line", conf.int = TRUE,
                  cor.coef = TRUE, cor.method = "pearson",
                  xlab = "Heart Rate", ylab = "Body Temperature")


rho <- cor(fdata$heart_rate,fdata$body_temperature) # finding Pearson's
correlation coeff.

rho

# found the point estimate of rho.

ggpubr::ggscatter(fdata, x = "heart_rate", y = "body_temperature",
                  add = "reg.line", conf.int = TRUE,
                  cor.coef = TRUE, cor.method = "pearson",
                  xlab = "Heart Rate", ylab = "Body Temperature")
```

Question 2:

a) Calculating Monte-Carlo estimates of coverage probabilities of the two intervals by simulating appropriate data.

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

```
> n <- 30
> lambda <- 0.1
> for (i in 1:5000){

    # Generating data following Exponential Distribution.
    data_sim <- rexp(n, lambda)

    # Calculating the Confidence Interval using Large Sample Mean formula.
    lower.z <- mean(data_sim) - (qnorm(.975)*sd(data_sim)/sqrt(n))
    upper.z <- mean(data_sim) + (qnorm(.975)*sd(data_sim)/sqrt(n))

    # Calculating Parametric Bootstrap Percentile CI.
    BootR <- boot(data=data_sim, statistic = mean.boot, R= 999)
    lower.boot <- quantile(BootR$t,.025)
    upper.boot <- quantile(BootR$t,.975)

    # Creating vectors for storing the conf intervals generated by large sample z-interval and parametric bootstrap method for each of the 5000 iterations.
    z_coverage[i] <- ((1/lambda) >= lower.z) & ((1/lambda) <= upper.z)
    boot_coverage[i] <- ((1/lambda) >= lower.boot) & ((1/lambda) <= upper.boot)
  }
```

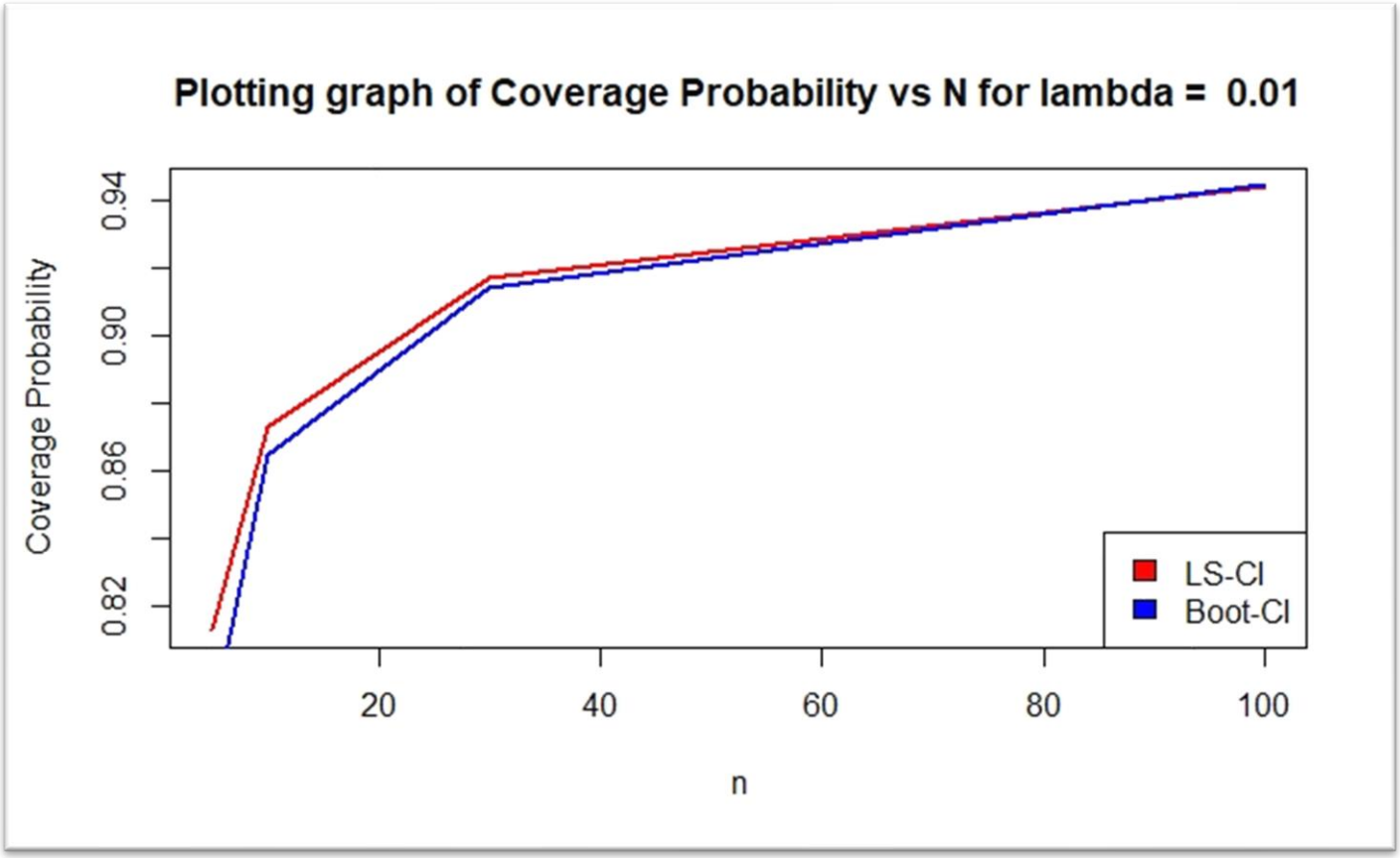
Results will be covered when displaying all (n, lambda) combinations.

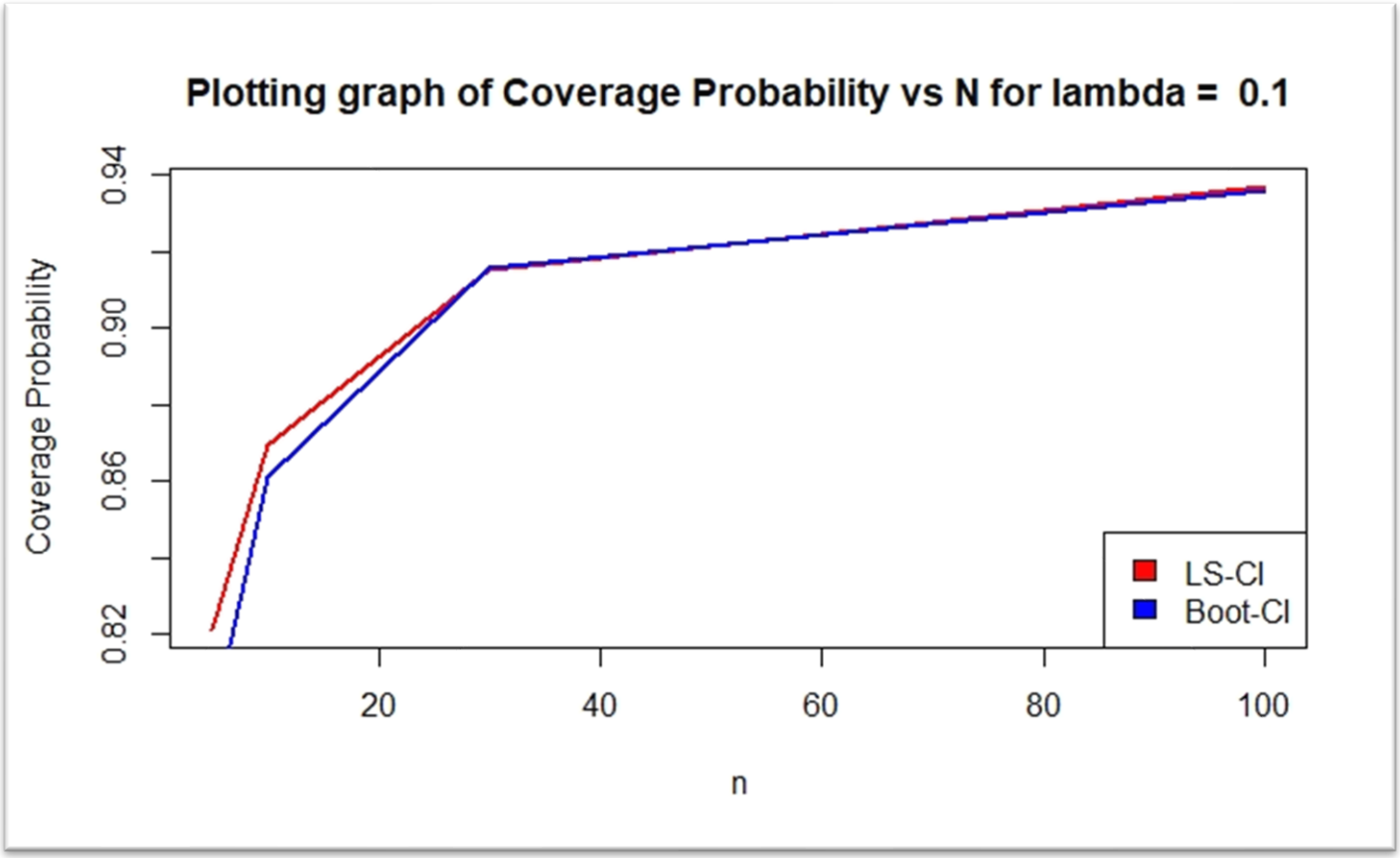
b) Now repeating the above process for all the combinations of {n, lambda}.

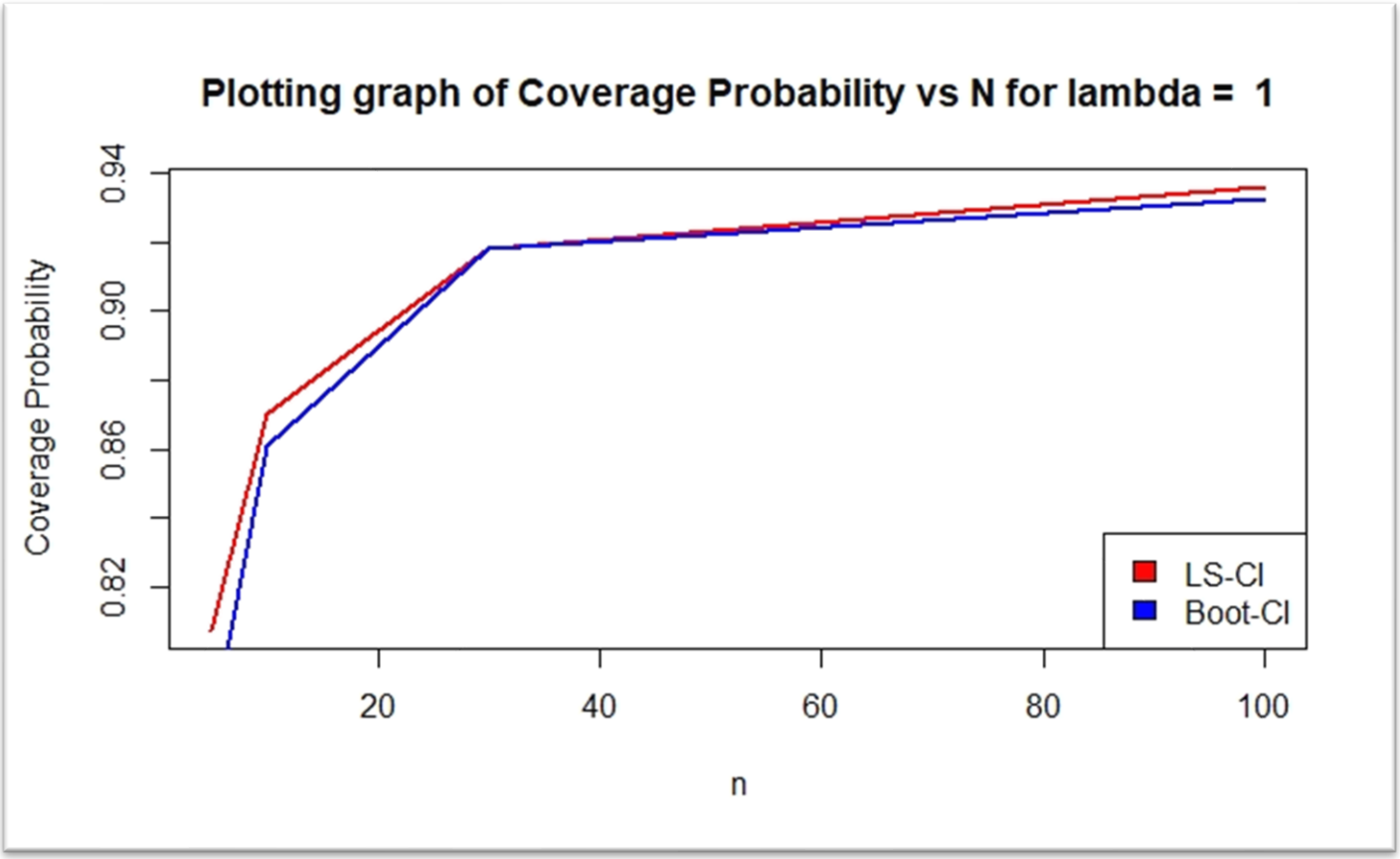
For this, we will automate the entire block of code shown earlier, by creating two vectors for storing values of n and lambda.
Then, keeping lambda constant, we iterate through all the 4 values of n, computing 5000 C.I. for each combination.

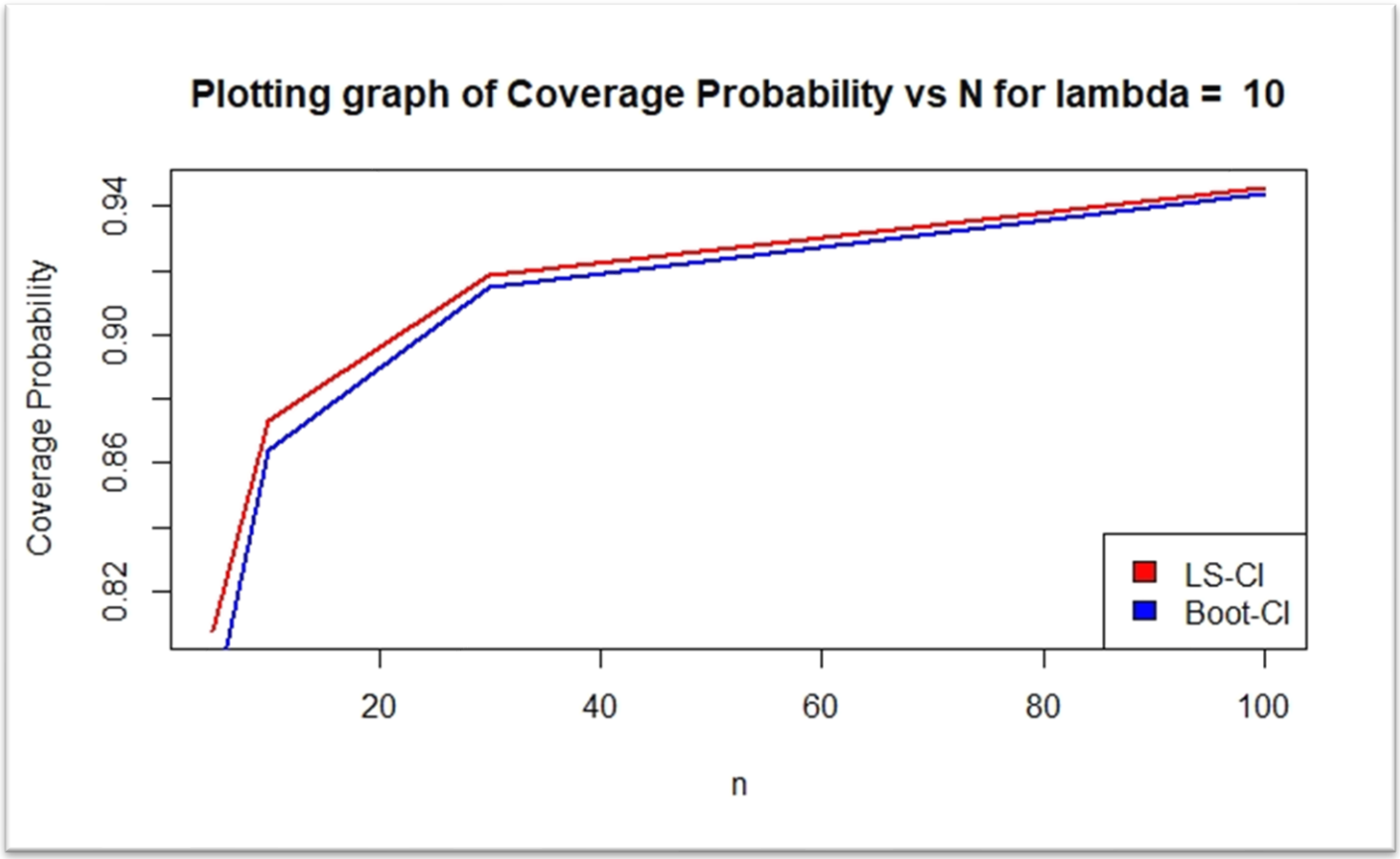
Finally, we plot these in the form of a line plot for ease of visualisation.

(The R Code will be provided at the end, so that the questions can be answered perfectly and concisely.)




Plotting graph of Coverage Probability vs N for lambda = 0.1
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 = 1
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, lambda} :

```
+ }
>
> data_info
      lamda   n z_coverage boot_coverage
 [1,]  0.01   5     0.8114         0.7854
 [2,]  0.01  10     0.8734         0.8616
 [3,]  0.01  30     0.9156         0.9152
 [4,]  0.01 100     0.9358         0.9338
 [5,]  0.10   5     0.8076         0.7824
 [6,]  0.10  10     0.8718         0.8648
 [7,]  0.10  30     0.9196         0.9182
 [8,]  0.10 100     0.9346         0.9350
 [9,]  1.00   5     0.8144         0.7872
[10,]  1.00  10     0.8690         0.8586
[11,]  1.00  30     0.9176         0.9166
[12,]  1.00 100     0.9390         0.9380
[13,] 10.00   5     0.8148         0.7838
[14,] 10.00  10     0.8668         0.8648
[15,] 10.00  30     0.9142         0.9146
[16,] 10.00 100     0.9414         0.9400
>
```

c) Interpreting the results of our computation:

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

A1) As we can see from our plots, for the Confidence Intervals generated by Large Sample z-test to be precise,
The sample size must be greater than 30 {n>30}.

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

A2) We know that Bootstrapping is a method which is preferred only when we have a sufficiently large dataset, n > 30, because only then will generating multiple samples from the original one through resampling techniques make sense!

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

A3) On observing the generated matrix of {n, lambda} we can see that the answers do not seem to depend on the value of lambda. Looking at the data, we cannot comment on the accuracy of either of the methods being better. However, z-test will generally prove to be more accurate if a large sample is given.

Q4) Which interval would you recommend?
A4) No recommendation can be made, as both are useful under their own conditions. The user must decide that by analysing the dataset that he has been provided with.

d) No, the conclusions reached in the above question are independent of the value of lambda parameter given to us, clearly proved by the table data.

R Code :

```
library(boot)
# Given: Investigation focuses on Confidence lvl (1-alpha) = 0.95
n <- c(5,10,30,100)
lambda <- c(0.01,0.1,1,10)

lower.z = numeric(0)
upper.z = numeric(0)
lower.boot = numeric(0)
upper.boot= numeric(0)
z_coverage = numeric(0)
boot_coverage = numeric(0)

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

data_info = matrix(,nrow = 1,ncol = 4,dimnames =
list(c(1),c("lamda","n","z_coverage","boot_coverage")))
data_info = data_info[-c(1),]
for(param in lambda){
  conf_lvl_LLN = numeric(0)
  conf_lvl_BootPerCI = numeric(0)
  k=1
  for(datasize in n){
    for (i in 1:5000){

      # Generating data following Exponential Distribution.
      data_sim <- rexp(datasize, param)

      # Calculating the Confidence Interval using Large Sample Mean
formula.
      lower.z <- mean(data_sim) - (qnorm(.975)*sd(data_sim)/sqrt(datasize))
      upper.z <- mean(data_sim) + (qnorm(.975)*sd(data_sim)/sqrt(datasize))
```

```
# Calculating Parametric Bootstrap Percentile CI.
      BootR <- boot(data=data_sim, statistic = mean.boot, R= 999)
      lower.boot <- quantile(BootR$t,.025)
      upper.boot <- quantile(BootR$t,.975)


      z_coverage[i] <- ((1/param) >= lower.z) & ((1/param) <= upper.z)
      boot_coverage[i] <- ((1/param) >= lower.boot) & ((1/param) <=
upper.boot)
    }


    conf_lvl_LLN[k] <- mean(z_coverage)
    conf_lvl_BootPerCI[k] <- mean(boot_coverage)


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


data_info
```