Statistical Methods for Data Science: Mini Project 5 Solved

Mini Project #: 5
Group #: 9
Names of group members: Nikita Ahuja, Bhargaw Rajnikant Patel
Contribution of each group member: Both team members worked together to solve the two problems. Both members reviewed the equations from the problems given, worked out the solutions in R, wrote the code and finished the report in a timely manner. Both partners worked equally to complete the Mini Project 5 requirements.

Problem 1:

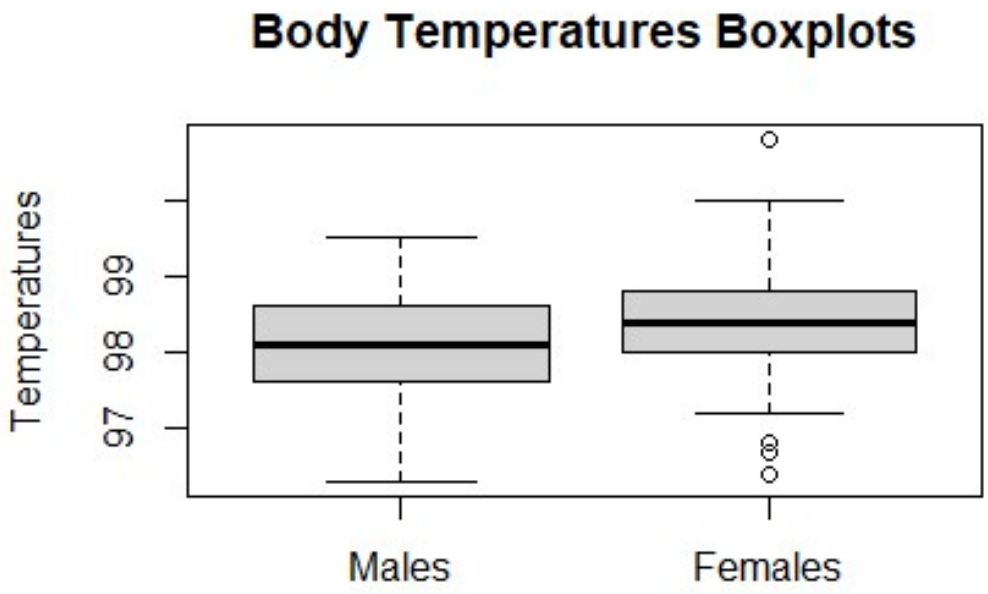
a. First read csv file using read.csv function. Then separate the two datasets using the subsets function, which returns a subset of vectors and matrices.

Draw the boxplots for the body temperature values for m=both females and males.

- **Code:**

```
> bodytemp_heartrate = read.csv("D:/UTD/Fall 21/6313 Stats/Mini Projects/Mini Project 5/bodytemp-heartrate.csv", header = T )
> males = subset(bodytemp_heartrate, bodytemp_heartrate$gender == 1)
> females = subset(bodytemp_heartrate, bodytemp_heartrate$gender == 2)
> boxplot(males$body_temperature, females$body_temperature, main = "Body Temperatures Boxplots", names = c('Males', 'Females'), ylab = "Temperatures")
```

- **Output:**



Observation: Because Q1, median, and Q3 are higher in females than in males, the female distribution may have a slightly higher mean than males. There are many outliers in female boxplots. This means that it is more volatile than men. Therefore, the same variance cannot be assumed.

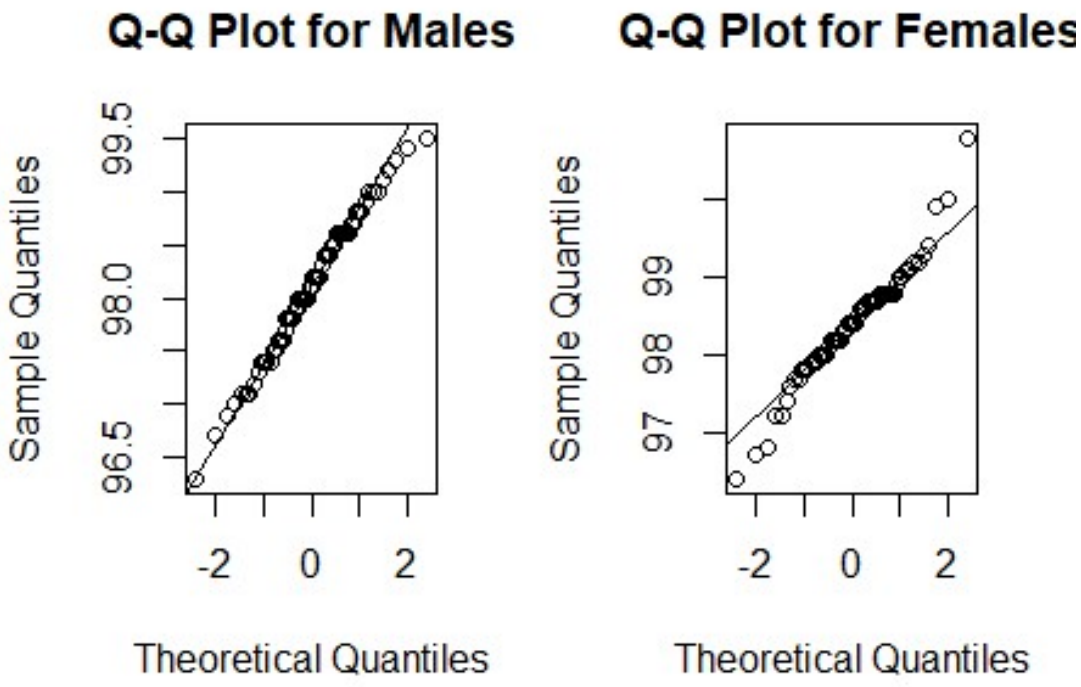
Draw Q-Q plot for these values

- **Code:**

```
> par(mfrow=c(1,2))
> qqnorm(males$body_temperature, main = 'Q-Q Plot for Males')
> qqline(males$body_temperature)
> qqnorm(females$body_temperature, main = 'Q-Q Plot for Females')
> qqline(females$body_temperature)
> t.test(males$body_temperature, females$body_temperature, alternative =
'two.sided', var.equal = F)

        Welch Two Sample t-test

data:  males$body_temperature and females$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

- **Output:**



Observations: As can be seen from the QQ plot, the distribution of of these temperature values for both men and women can be considered to be a normal value of about.

We take the null hypothesis $H0$: means difference = 0 => $m_m - f = 0$
And Alternate Hypothesis H_1 : means difference≠0 => $m_f - f \neq 0$

Were mm will estimate population mean for male and mf will estimate population for female.

Treat the sample here as an independent sample. Since the unequal distribution is derived from the approximate normal distribution, you can use t-distribution in the Satterthwaite approximation to get the confidence intervals.

Construct the confidence interval using t.test function in R.

The confidence interval we observe as a result of the function t.test in R is (-0.53964856, -0.03881298)
The p-value we got is 0.02394.

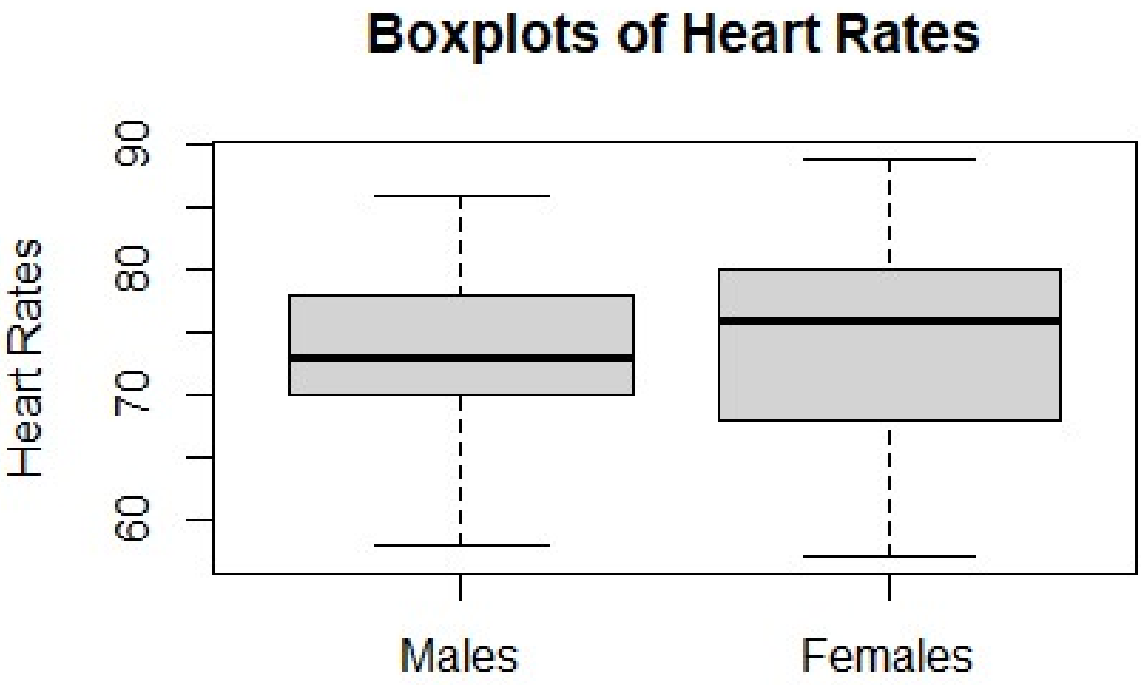
Since the p-value is less than 0.05 and 0 is not in the confidence interval, we reject the null hypothesis and therefore conclude that the mean temperature of females and males is not the same. The width of the confidence interval is so small that the sample mean will differ for very small values. And the average female body temperature is slightly higher than its counterpart.

b. draw the boxplots for the heart rate values for both females and males

- **Code:**

```
> boxplot(males$heart_rate, females$heart_rate, main = "Boxplots of Heart Rates",names = c('Males', 'Females'), ylab = "Heart Rates")
```

- **Output:**



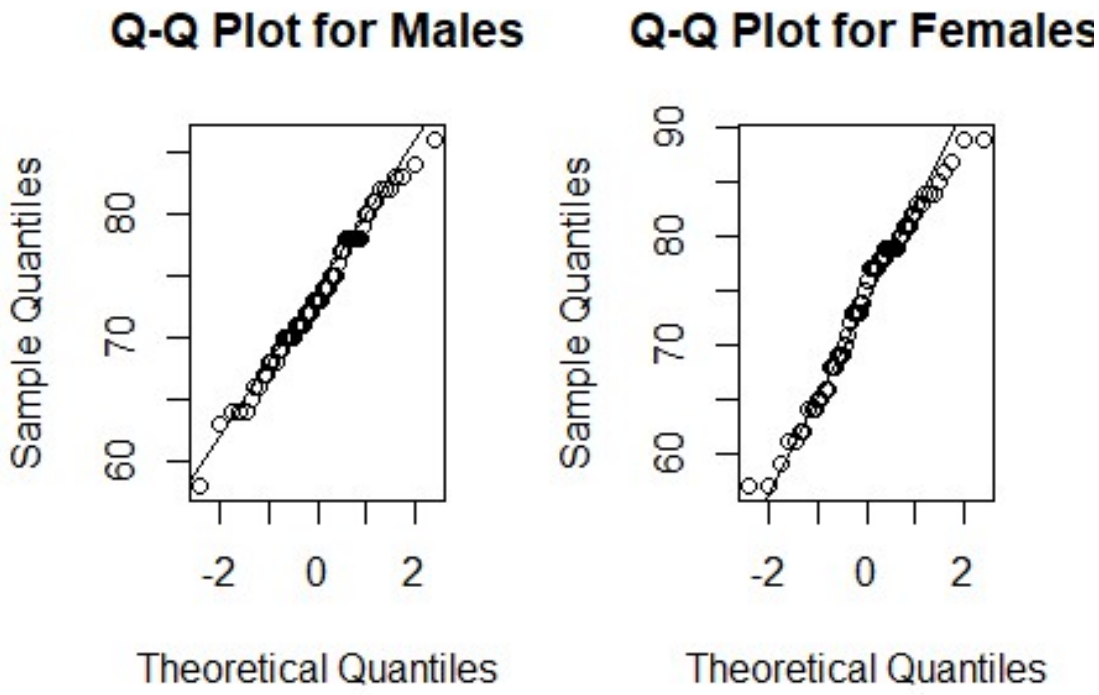
Observation: Female Q1 is smaller than male Q1, but these values are higher in females than in males, so they do not fit the median of and Q3. The female value appears to be more stretched, so it appears to be more volatile.

draw Q-Q plot for these values

- **Code:**

```
> par(mfrow=c(1,2))
> qqnorm(males$heart_rate, main = 'Q-Q Plot for Males')
> qqline(males$heart_rate)
> qqnorm(females$heart_rate, main = 'Q-Q Plot for Females')
> qqline(females$heart_rate)
> t.test(males$heart_rate, females$heart_rate, alternative = 'two.sided', var.equal = F)

        Welch Two Sample t-test

data:  males$heart_rate and females$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
```

- **Output:**



Observations: As can be seen from the QQ plot, the distribution of these heart rate measurements can be nearly normal in both men and women.

We take the null hypothesis $H0$: means difference = 0 => $m_m - f = 0$
And Alternate Hypothesis H_1 : means difference≠0 => $m_f - f \neq 0$
Were mm will estimate population mean for male and mf will estimate population for female.

Since the sample here is treated as an independent sample, and the uneven variance comes from a nearly normal distribution, we can use t-distribution with Satterthwaite's approximation to get the confidence interval

construct the confidence interval using t.test function in R
The confidence interval we observe as a result of the function t.test in R is
(-3.243732, 1.674501)
The p-value we got is 0.5287.

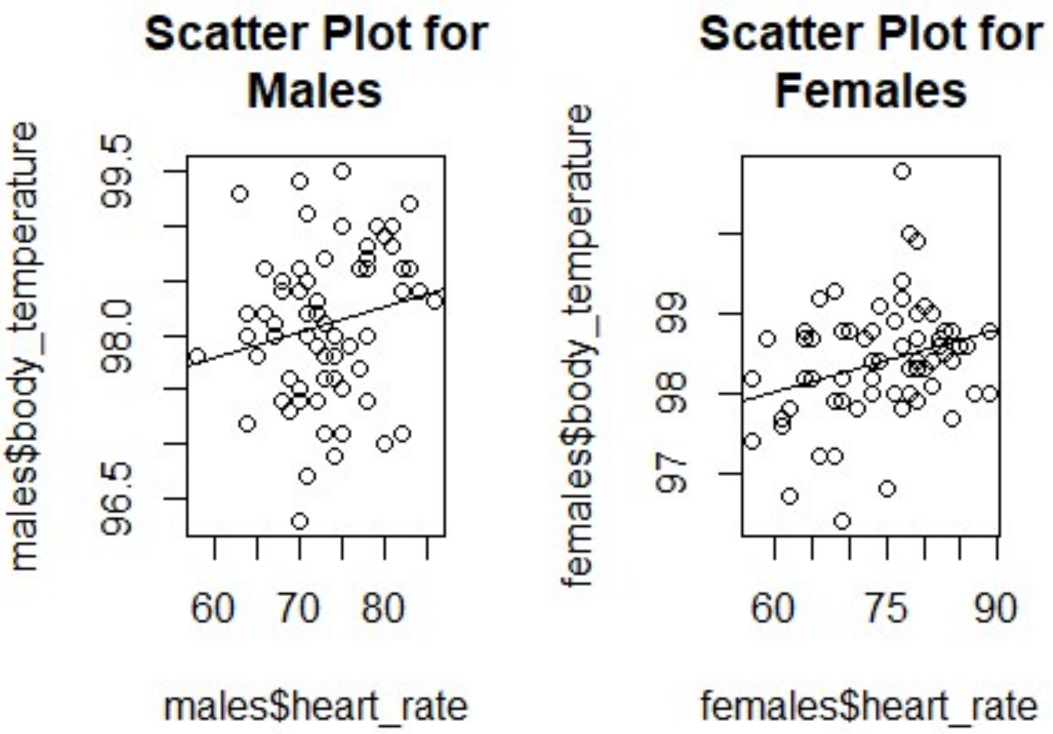
Since the p-value is greater than 0.05 and the value 0 is in the confidence interval , we accept the null hypothesis and therefore come to the conclusion that the mean heart rate is the same for females and males.

c. Draw a scatter plot and then draw a regression line that reflects the linear relationship between them.

- **Code:**

```
> par(mfrow=c(1,2))
> plot(males$heart_rate, males$body_temperature, pch=1, main='Scatter Plot for
+ Males')
> abline(lm(males$body_temperature~males$heart_rate))
> plot(females$heart_rate, females$body_temperature, pch=1, main='Scatter
Plot for
+ Females')
> abline(lm(females$body_temperature~females$heart_rate))
```

- **Output:**



Observation: As you can see from the graph, the slope of the drawn line is larger than 0. This suggests that there is a positive correlation between body temperature and heart rate readings. From the graph, it can be inferred that the intensity of the linear relationship is weak.

You can now use the function cor to determine the correlation between the two variables.

```
cor(females$body_temperature,females$heart_rate)
[1] 0.2869312
```

```
cor(males$body_temperature,males$heart_rate)
[1] 0.1955894
```

Based on the given data we get
Correlation between body temperature and heart rate for males is: 0.1955894
Correlation between body temperature and heart rate for females is: 0.2869312

We know that the higher the value, the stronger the correlation, so we conclude here that the relationship between body temperature and heart rate is weak. In the case of, the correlation between body temperature and heart rate is slightly stronger in females than in males, because the correlation is higher in females than in males.

Problem 2: a. We have to simulate Monte Carlo estimates of coverage probabilities of the two intervals by simulating appropriate data. We have to construct two confidence intervals and repeat the process 5000 times.

1. checkzci – It takes n and λ values as the input parameters, simulates the sample, constructs an interval and returns whether the true mean exists within the confidence interval.
2. zproportion – It takes n and λ vales as the input parameters, calls the checkzci function 5000 times and calculates the coverage probabilities.
3. mean_cal – Since we have to form samples from the distribution, this function returns the mean.
4. checkbci – It takes n and λ given as the input parameters, calls the mean_cal function 1000 times and forms the confidence interval. It then returns whether the true mean is present in the interval.
5. bproportion – It takes n and λ as the input parameters, constructs the parametric initial bootstrap sample and calls checkbci 5000 times. This function calculates the coverage probabilities.

```
> 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)
+ }

> # n = 5 and lambda = 0.01 for zproportion
> zproportion(5,0.01)
[1] 0.8074

> mean_cal <- function(n,lambda) {
+     u_cal <- rexp(n, lambda)
+     return (mean(u_cal))
+ }
```

```
> checkbci <- function(n, lambda) {
+     U <- rexp(n,lambda)
+     tm <- 1/lambda
+     lambda1 = 1/mean(U)
+    V <- replicate(1000, mean_cal(n,lambda1))
+    bound <- sort(V)[c(25, 975)]
+    if(bound[2]>tm & bound[1]<tm) {
+      return (1)
+      }
+     else {
+      return (0)
+      }
+  }
> bproportion <- function(n, lambda) {
+     values <- replicate(5000, checkbci(n, lambda))
+     ones <- values[which (values == 1)]
+     return (length(ones)/5000)
+ }

> bproportion(5,0.01)
[1] 0.8976
```

Using these functions, for the (n,λ) combination as (5, 0.01) we get the coverage probabilities as:
Z-interval: 0.8074
Bootstrap interval: 0.8976

b. Repeating the process for the remaining combinations, we have :-

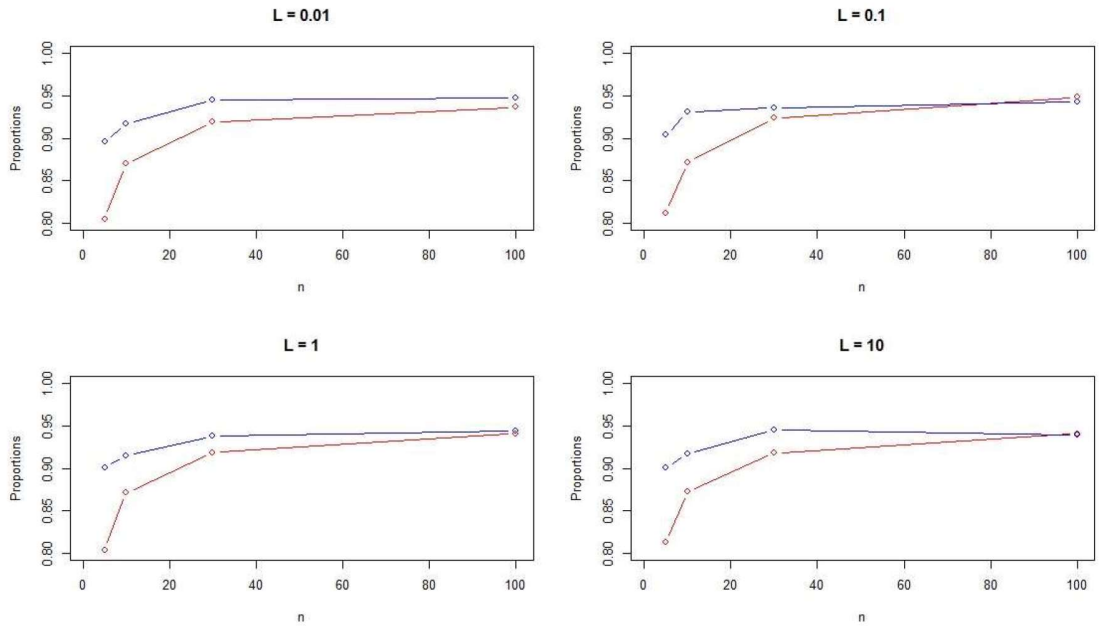
Z proportions	L = 0.01	L = 0.1	L = 1	L = 10
N = 5	0.8074	0.8134	0.8178	0.8102
N = 10	0.8712	0.8710	0.8736	0.8708
N = 30	0.9092	0.9218	0.9162	0.9188
N = 100	0.9364	0.9426	0.9406	0.9452

B proportions	L = 0.01	L = 0.1	L = 1	L = 10
N = 5	0.8976	0.9034	0.9003	0.9056
N = 10	0.9158	0.9334	0.9158	0.9163
N = 30	0.9452	0.9365	0.9342	0.9482
N = 100	0.9478	0.9428	0.9492	0.9361

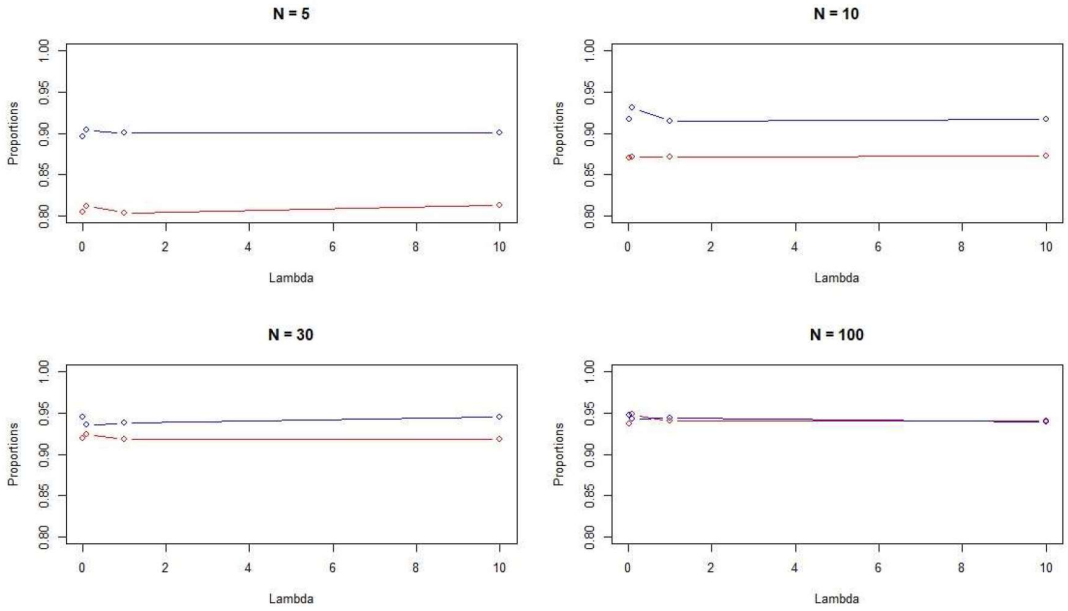
Now, we generate the matrices of the proportion values for the Z interval and Bootstrap for all combinations of N & L. We then plot the same.

```
> zcimatrix <- matrix(c(zproportion(5,0.01), zproportion(10,0.01),
+ zproportion(30,0.01), zproportion(100,0.01), zproportion(5,0.1), zproportion
(10,0.1),zproportion(30,0.1), zproportion(100,0.1), zproportion(5,1), zproporti
on(10,1),
+ zproportion(30,1), zproportion(100,1), zproportion(5,10), zproportion(10,10),
+ zproportion(30,10), zproportion(100,10)), nrow = 4, ncol = 4)
>
> bcimatrix <- matrix(c(bproportion(5,0.01), bproportion(10,0.01),
+                       bproportion(30,0.01), bproportion(100,0.01), bproportio
n(5,0.1), bproportion(10,0.1),bproportion(30,0.1), bproportion(100,0.1), bpropo
rtion(5,1), bproportion(10,1),bproportion(30,1), bproportion(100,1), bproportio
n(5,10), bproportion(10,10),bproportion(30,10), bproportion(100,10)), nrow = 4,
 ncol = 4)
```

```
> par(mfrow=c(2,2))
> plot(c(5,10,30,100), zcimatrix[,1], main = "L = 0.01", xlab = 'n', ylab ='Proportions', col = 'red',
type = 'b', xlim = c(1,100), ylim = c(0,1))
> lines(c(5,10,30,100), bcimatrix[,1], col = 'blue', type = 'b')

> plot(c(5,10,30,100), zcimatrix[,2], main = "L = 0.1", xlab = 'n', ylab = 'Proportions', col = 'red',
type = 'b', xlim = c(1,100), ylim = c(0,1))
> lines(c(5,10,30,100), bcimatrix[,2], col = 'blue', type = 'b')

> plot(c(5,10,30,100), zcimatrix[,3], main = "L = 1", xlab = 'n', ylab = 'Proportions', col = 'red',
type = 'b', xlim = c(1,100), ylim = c(0,1))
> lines(c(5,10,30,100), bcimatrix[,3], col = 'blue', type = 'b')

> plot(c(5,10,30,100), zcimatrix[,4], main = "L = 10", xlab = 'n', ylab = 'Proportions', col = 'red',
type = 'b', xlim = c(1,100), ylim = c(0,1))
> lines(c(5,10,30,100), bcimatrix[,4], col = 'blue', type = 'b')

> plot(c(0.01,0.1,1,10), zcimatrix[1,], main = "N = 5", xlab = 'Lambda', ylab = 'Proportions', col =
'red', type = 'b', xlim = c(0.01,10), ylim = c(0,1))
> lines(c(0.01,0.1,1,10), bcimatrix[1,], col = 'blue', type = 'b')

> plot(c(0.01,0.1,1,10), zcimatrix[2,], main = "N = 10", xlab = 'Lambda', ylab = 'Proportions', col =
'red', type = 'b', xlim = c(0.01,10), ylim = c(0,1))
> lines(c(0.01,0.1,1,10), bcimatrix[2,], col = 'blue', type = 'b')

> plot(c(0.01,0.1,1,10), zcimatrix[3,], main = "N = 30", xlab = 'Lambda', ylab = 'Proportions', col =
'red', type = 'b', xlim = c(0.01,10), ylim = c(0,1))
> lines(c(0.01,0.1,1,10), bcimatrix[3,], col = 'blue', type = 'b')

> plot(c(0.01,0.1,1,10), zcimatrix[4,], main = "N = 100", xlab = 'Lambda', ylab ='Proportions', col =
'red', type = 'b', xlim = c(0.01,10), ylim = c(0,1))
> lines(c(0.01,0.1,1,10), bcimatrix[4,], col = 'blue', type = 'b')
```



GRAPHS A: Here, the red line denotes the z proportions and the blue line denotes the bootstrap proportions. The values are plotted for n while keep lambda fixed.



GRAPHS B: Here, the red line denotes the z proportions and the blue line denotes the bootstrap proportions. The values are plotted for lambda while keep n fixed.

c. From GRAPHS A, we can see that the graphs don't change drastically when λ is changed, and therefore we can say that the coverage probabilities don't depend on λ. We also see that the coverage probabilities we get via bootstrap are higher than those of z-interval method.

From GRAPHS B, 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).

The coverage probabilities for the bootstrap method are on the higher side (approximately) from n=30 onwards.

From 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. Therefore, the bootstrap method is more accurate even for the low values of n and we can choose this method for our problem.

d. The output from the code in Section 2 helps us to infer that the:
The coverage probability for bootstrap is for n = 5 lambda = 0.1 is 0.6.
The coverage probability for large sample z for n = 5 lambda = 0.1 is 0.8.
And,
The coverage probability for bootstrap is for n = 10 lambda = 0.1 is 0.6.
The coverage probability for large sample z for n = 10 lambda = 0.1 is 0.8.
The coverage probability for bootstrap is for n = 30 lambda = 0.1 is 0.7.
The coverage probability for large sample z for n = 30 lambda 0.1 is 0.9.
The coverage probability for bootstrap is for n = 100 lambda = 0.1 is 0.7.
The coverage probability for large sample z for n = 100 lambda = 0.1 is 0.9.

Therefore, the conclusions obtained in (c) hold for specific values of lambda. In this case lambda = 0.1.