**Mini Project # 5**

**Names of group members: (Group-18)**

Deepika Mamidipelly

Preethi Pasunuri

**Contribution of each group member**

Both group members contributed equally to the inputs for both questions and best is chosen among them to solve these problems. Collaboratively learned R, ran the scripts, and assessed the results. Some of the scripts written by Deepika were analysed and finalized by Preethi and similarly scripts written by Preethi were assessed and finalized by Deepika. Both group members distribute equal amounts of report documentation, which is then integrated into a single final document. Members of the group worked diligently to meet all the project criteria.

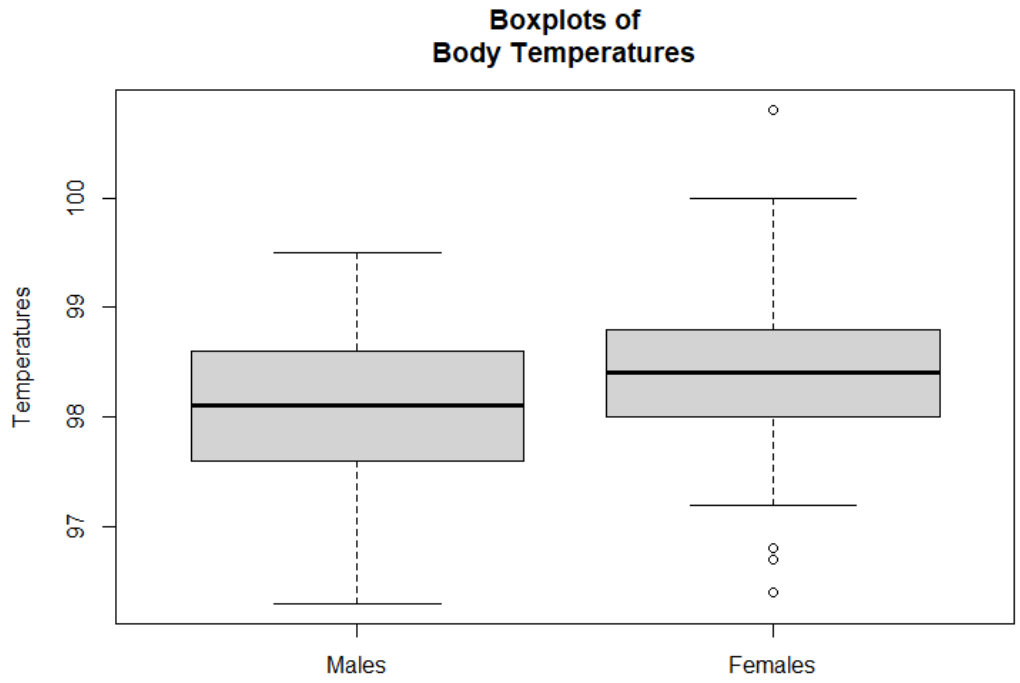
**Question 1:**

1) **Solution:**

Firstly we use the read.csv function to read the csv data into the variable.
Now separate the two data sets by using the subset function which returns subsets of vectors and matrices.
We can follow this data set with any relational condition we need to separate the data set.

**a) Solution:**

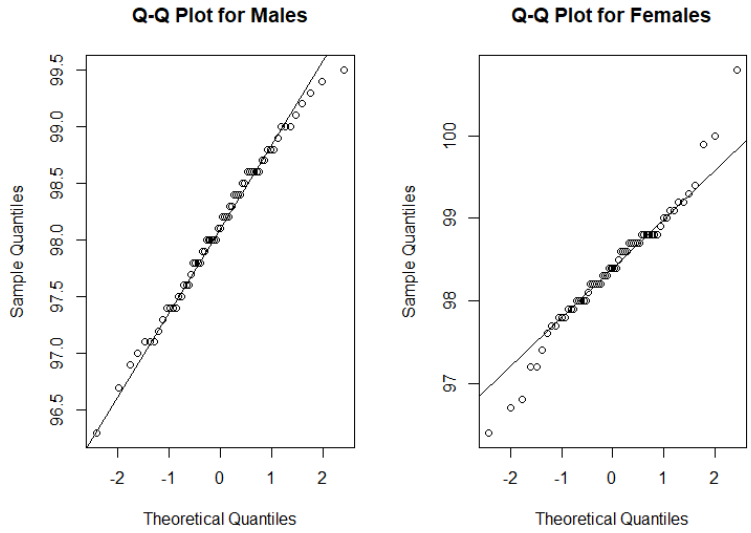
Let's draw the boxplots for the body temperature values for m=both females and males.




**Observations:** 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.

There are more outliers in the females' box plot implies there more variability for them than the males. Hence, we cannot assume equal variances.

**Let's draw Q-Q plot for these values:**




**Observations:** 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.

Let 'M' denote the body temperatures of males and 'F' denote the body temperatures of females.

So the sample mean $m^-$ estimates the population mean $\mu_m$ and the sample mean $\bar{f}$ estimates the population mean $\mu_f$

We take the null hypothesis $H_0$ : Difference between means = 0 => $m^- - \bar{f} = 0$

And Alternate Hypothesis $H_1$ : Difference between means $\neq 0$ => $m^- - \bar{f} \neq 0$

The samples here are to be treated as independent samples, with unequal variances coming from an approximately normal distribution, hence we can use t-distribution with Satterthwaite's approximation to get the confidence interval.

We construct the confidence interval using **t.test** function in R

```
+  two.sided , var.equal = F)

        Welch Two Sample t-test

data:  maleValues$body_temperature and femaleValues$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
```

The confidence interval we observe as a result of the function t.test in R is

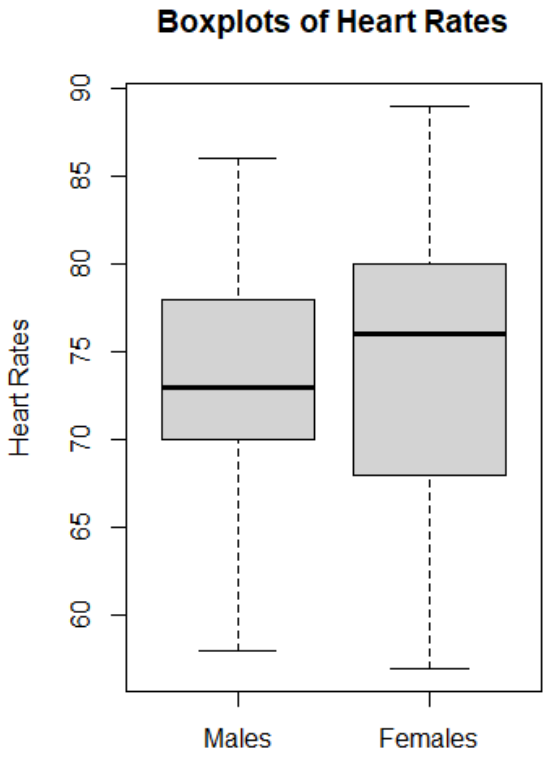
**(-0.53964856, -0.03881298 )**

The p-value we got is **0.02394**

Since p-value is less than 0.05 and 0 does not lie in the confidence interval, we reject the null hypothesis and hence come to the conclusion 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)Solution**

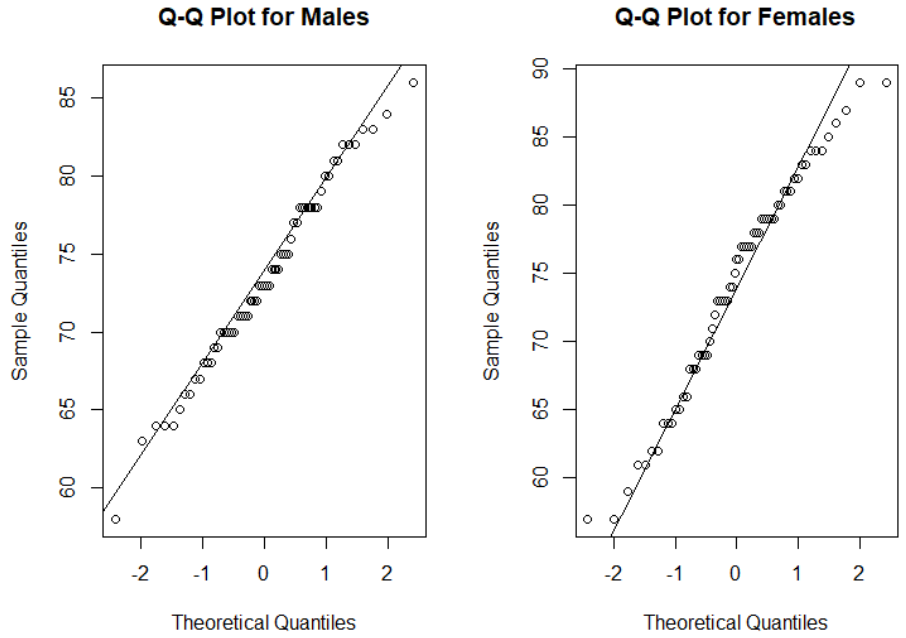
Let’s draw the boxplots for the heart rate values for both females and males:




**Observations:** 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.

**Lets draw Q-Q plot for these values:**




**Observations:** 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.

Let 'M' denote the body temperatures of males and 'F' denote the body temperatures of females.

So the sample mean $m^-$ estimates the population mean $\mu_m$ and the sample mean $f^-$

estimates the population mean $\mu_f$

We take the null hypothesis $H_0$: Difference between means = 0 => $m^- - f^- = 0$

And Alternate Hypothesis $H_1$ : Difference between means $\neq 0$ => $m^- - f^- \neq 0$

The samples here are to be treated as independent samples, with unequal variances coming from an approximately normal distribution, hence we can use **t-distribution** with Satterthwaite's approximation to get the confidence interval.

We construct the confidence interval using **t.test** function in R.

```
> t.test(maleValues$heart_rate, femaleValues$heart_rate, alternative = "two.sided", var.equ

        Welch Two Sample t-test

data:  maleValues$heart_rate and femaleValues$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

> |
```

The confidence interval we observe as a result of the function **t.test** in R is

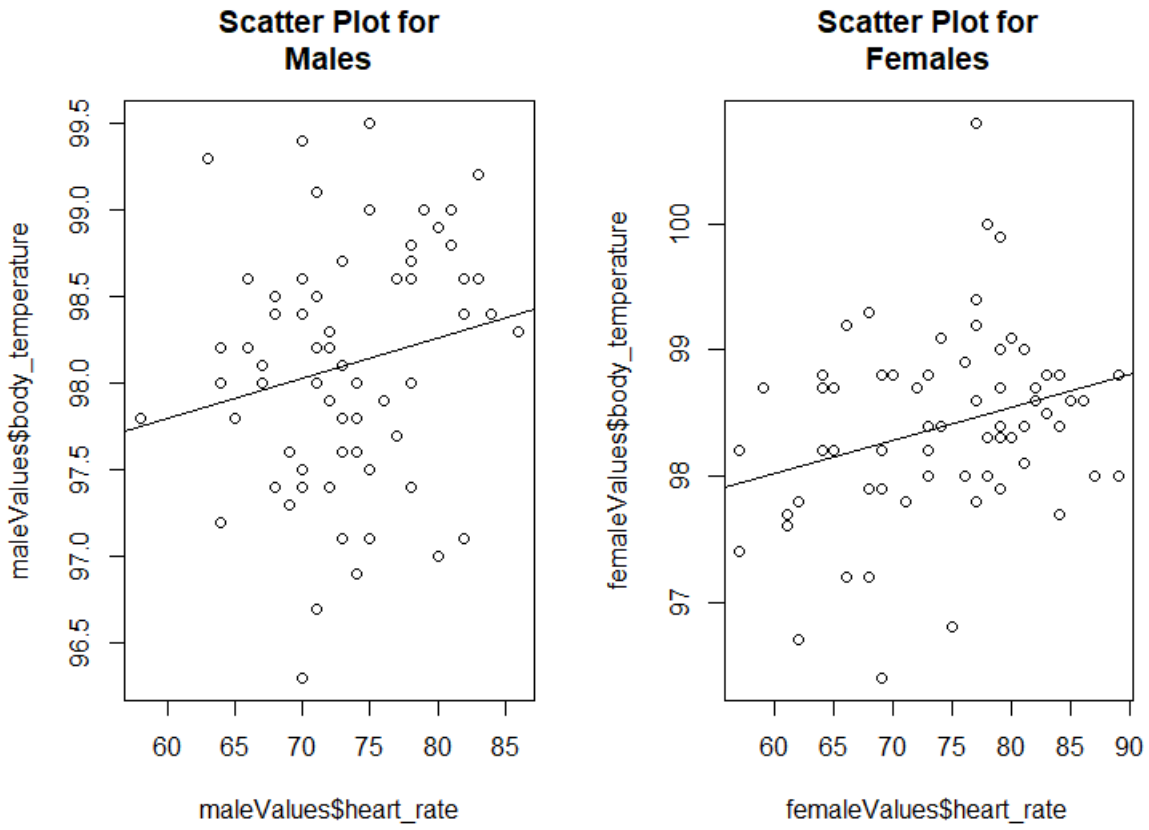
**( -3.243732, 1.674501 )**

The p-value we got is **0.5287.**

Since p-value is greater than 0.05 and the value 0 lies in the confidence interval, we accept the null hypothesis and hence come to the conclusion that the heart rate value means of females and males are equal.

**C. solution)**

Let us draw and consider a scatter plot and further plot a regression line that reflects the linear relationship between them.




**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.

Now, we can get the correlation between two variables by using the function **cor.**

Based on the given data we get

```
> #finding the correlation values between body temperatures and heart rates
> cor( maleValues$body_temperature,maleValues$heart_rate)
[1] 0.1955894
> cor( femaleValues$body_temperature,femaleValues$heart_rate)
[1] 0.2869312
>
```

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.

**R-Code:**

```
bodytemp_heartrateValues=read.csv("D:\\classes\\Statistics\\Assignments\\bodytemp-heartrate.csv", header = T )

maleValues = subset(bodytemp_heartrateValues, bodytemp_heartrateValues$gender == 1)

femaleValues = subset(bodytemp_heartrateValues,bodytemp_heartrateValues$gender == 2)

boxplot(maleValues$body_temperature, femaleValues$body_temperature, main = "Boxplots of Body Temperatures", names = c('Males', 'Females'), ylab = "Temperatures")

par(mfrow=c(1,2))

qqnorm(maleValues$body_temperature, main = 'Q-Q Plot for Males')

qqline(maleValues$body_temperature)

qqnorm(femaleValues$body_temperature, main = 'Q-Q Plot for Females')

qqline(femaleValues$body_temperature)
```

**#confidence interval using t,test function for the body temperature values**

```
t.test(maleValues$body_temperature, femaleValues$body_temperature, alternative =

'two.sided', var.equal = F)

boxplot(maleValues$heart_rate, femaleValues$heart_rate, main = "Boxplots of Heart Rates",

names = c('Males', 'Females'), ylab = "Heart Rates")
```

**#drawing Q-Q plot for the heart rate values**

```
par(mfrow=c(1,2))

qqnorm(maleValues$heart_rate, main = 'Q-Q Plot for Males')

qqline(maleValues$heart_rate)

qqnorm(femaleValues$heart_rate, main = 'Q-Q Plot for Females')

qqline(femaleValues$heart_rate)
```

**#getting the confidence interval using the t.test function**

```
t.test(maleValues$heart_rate, femaleValues$heart_rate, alternative = 'two.sided', var.equal = F)
```

**#finding the correlation values between body temperatures and heart rates**

```
cor( maleValues$body_temperature,maleValues$heart_rate)

cor( femaleValues$body_temperature,femaleValues$heart_rate)
```

**#drawing the scatter plots for the body temperature and heart rate values for males and females**

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

**Question-2)**

**Solution:**

**a)** To simulate Monte Carlo estimates of coverage probabilities and to construct the confidence intervals, we have created the following functions:

checkzci – takes n and $\lambda$ 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 $\lambda$ vales as input parameters, calls the checkzci unction 5000 times and calculates the coverage probabilities mean.star – samples from a distribution and returns the mean

checkbci – using the n and $\lambda$ 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 $\lambda$ as input parameters, constructs a parametric initial bootstrap sample and calls checkbci 5000 times and calculates the coverage probabilities

Using these functions, for the (n,$\lambda$) combination as (5, 0.01) we get the coverage probabilities as:

Z-interval: 0.8062

Bootstrap interval: 0.9008.

**RCode:**

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

>  #getting the value of n = 5 and lambda = 0.01 for zproportion
> zproportion(5,0.01)
[1] 0.8062
>
> #creating function mean.star
> mean.star <- function(n,lambda) {
+   u.star <- rexp(n, lambda)
+   return (mean(u.star))
+   }

> #creating function checkbci
> 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)
+     }
+   }
```

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

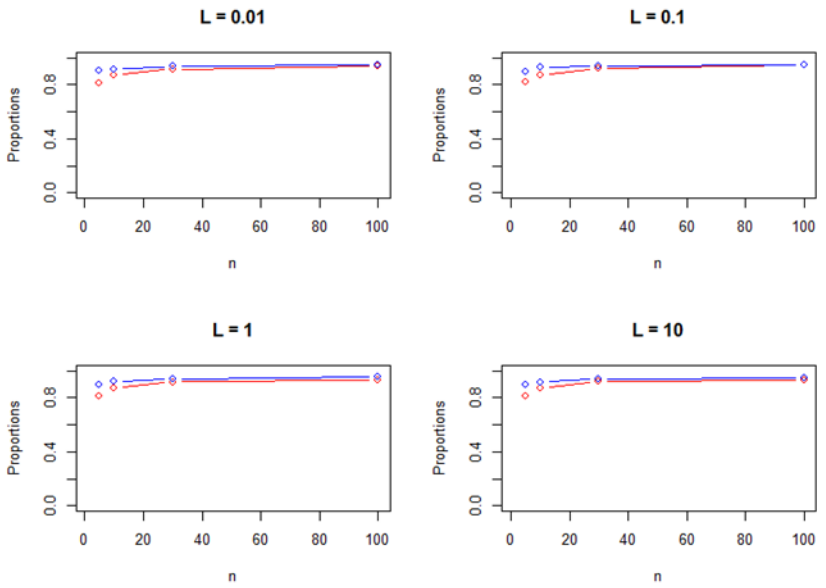
```
> # getting the value of n = 5 and lambda = 0.01 for bproportion
> bproportion(5,0.01)
[1] 0.9008
```

**b)** Repeating the above process for the remaining combinations we get:

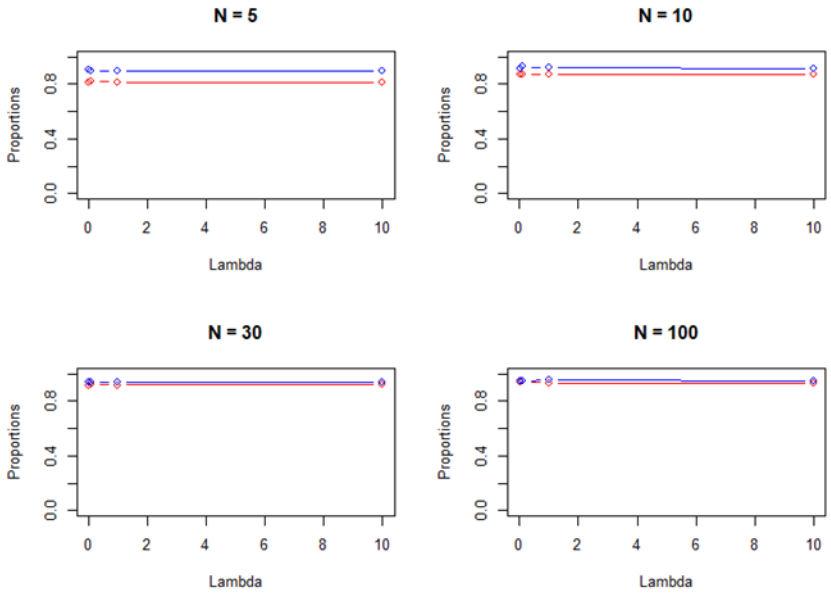
| Z-proportions | L = 0.01 | L = 0.1 | L = 1 | L = 10 |
|---|---|---|---|---|
| N = 5 | 0.8056 | 0.8124 | 0.8042 | 0.8132 |
| N = 10 | 0.8702 | 0.8716 | 0.8716 | 0.8728 |
| N = 30 | 0.9192 | 0.9236 | 0.9184 | 0.9178 |
| N = 100 | 0.9366 | 0.9482 | 0.9404 | 0.9408 |

| B-proportions | L = 0.01 | L = 0.1 | L = 1 | L = 10 |
|---|---|---|---|---|
| N = 5 | 0.8960 | 0.9038 | 0.9004 | 0.9002 |
| N = 10 | 0.9168 | 0.9304 | 0.9148 | 0.9172 |
| N = 30 | 0.9452 | 0.9356 | 0.9374 | 0.9448 |
| N = 100 | 0.9478 | 0.9430 | 0.9434 | 0.9388 |

Graphically representing the data, we get




**Graph 1**: Red represents z-proportions and blue represents bootstrap proportions. The values are plotted against n keeping $\lambda$ fixed




**Graph 2:** Red represents z-proportions and blue represents bootstrap proportions. The values are plotted against $\lambda$ keeping n fixed.

**RCode:**

**#getting the value of n = 5 and lambda = 0.01 for zproportion**

```
zproportion(5,0.01)

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), zproportion(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), bproportion(5,0.1), bproportion(10,0.1),
bproportion(30,0.1), bproportion(100,0.1), bproportion(5,1), bproportion(10,1),
bproportion(30,1), bproportion(100,1), bproportion(5,10), bproportion(10,10),
bproportion(30,10), bproportion(100,10)), nrow = 4, ncol = 4)
```

**# drawing line graphs for all these values**

```
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')
```

**c)**We can see from Graph 1, that the graphs do not drastically change when $\lambda$ is changed, so we can conclude that the coverage probabilities are independent of $\lambda$. We can also see that the coverage probabilities obtained by the bootstrap method are greater than those obtained by the z-interval method

We can deduce from Graph 2 that the coverage probabilities are proportional to n. Now comes the interesting part. We can see from the large-sample z-interval that the coverage probabilities are as accurate as the coverage. When n is large (n=100), we get probabilities from the bootstrap method. From n=30 onwards, the bootstrap method coverage probabilities are on the higher side (approximately). Taking into account all of the graphs, we can conclude that the coverage probabilities obtained from the bootstrap method are higher for every combination of (n,$\lambda$) than those obtained from the large-sample z-interval method, implying that the bootstrap method is more accurate even for low values of n. As a result, the bootstrap method is recommended.

**d)**The output from the code helps us to infer that the:

The coverage probability for bootstrap is for n =5 lambda =0.1 is 0.61

The coverage probability for large sample z for n =5 lambda =0.1 is 0.8058

The coverage probability for bootstrap is for n =10 lambda= 0.1 is 0.695 .

The coverage probability for large sample z for n= 10 lambda =0.1 is 0.8758.

The coverage probability for bootstrap is for n 30 lambda 0.1 is 0.7134 .

The coverage probability for large sample z for n 30 lambda 0.1 is 0.9114.

The coverage probability for bootstrap is for n 100 lambda 0.1 is 0.7218.

The coverage probability for large sample z for n 100 lambda 0.1 is 0.9388.

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