STATISTICAL METHODS FOR DATA SCIENCE - MINI PROJECT 5

Names of group members: 1. Venkatesh Sankar Net ID: VXS200014
2. Manneyaa Jayasanker Net ID: MXJ180040

Contribution of each group member:

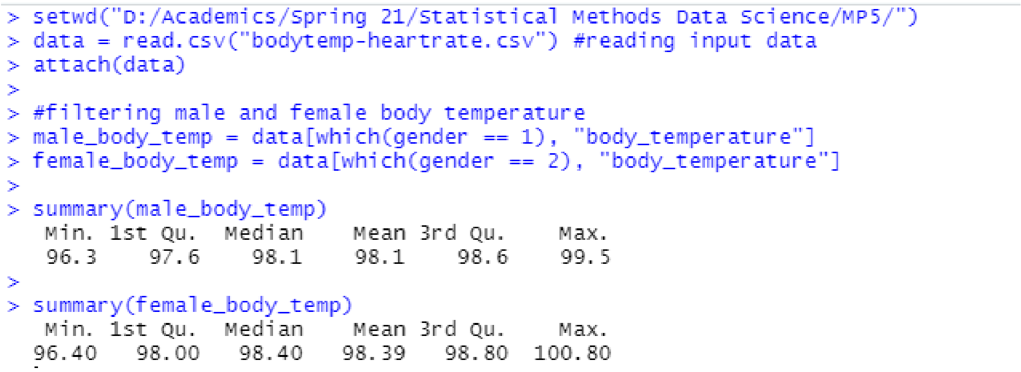
Venkatesh Sankar:
- Worked on R code for 1 a), 1 b) and 1 c).
- Worked on conclusion for 1 a), 1 b) and 1 c).
- Wrote documentation for 1 a), 1 b) and 1 c).

Manneyaa Jayasanker:
- Worked on R code for 2 a), 2 b) and 2 c) and 2 d).
- Worked on conclusion for 2 a), 2 b) and 2 c) and 2 d).
- Wrote documentation for 2 a), 2 b) and 2 c) and 2 d).

===

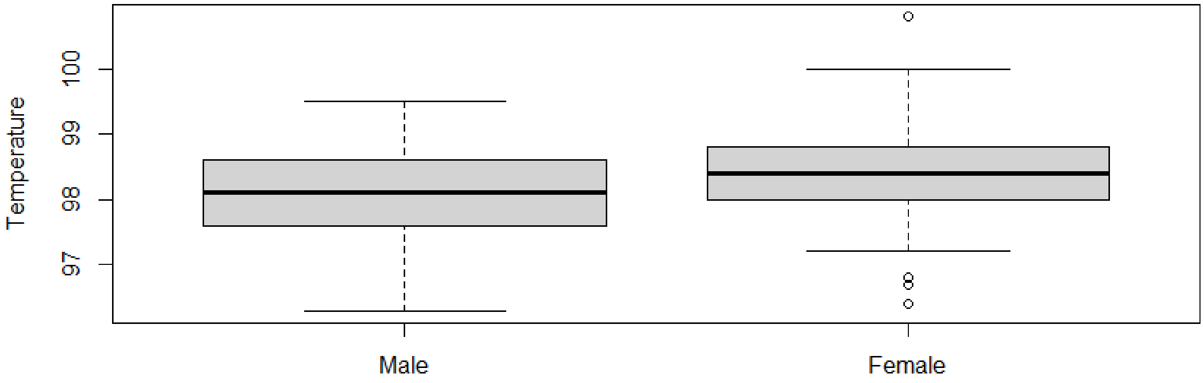
1.

a) We filter the male and female body temperature based on ‘gender’ column from the given input data. Given below is the summary of male body temperature and female body temperature.

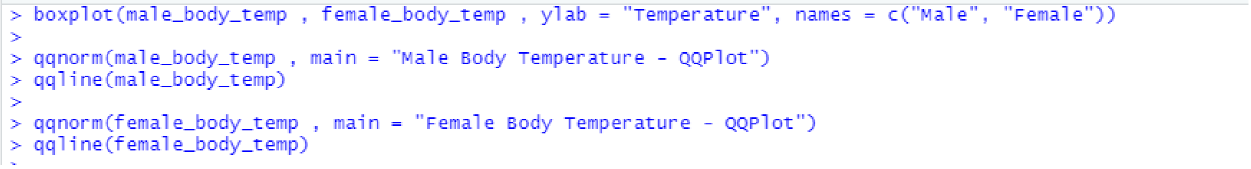


```
> setwd("D:/Academics/Spring 21/Statistical Methods Data Science/MP5/")
> data = read.csv("bodytemp-heartrate.csv") #reading input data
> attach(data)
>
> #filtering male and female body temperature
> male_body_temp = data[which(gender == 1), "body_temperature"]
> female_body_temp = data[which(gender == 2), "body_temperature"]
>
> summary(male_body_temp)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   96.3    97.6    98.1    98.1    98.6    99.5 
>
> summary(female_body_temp)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  96.40   98.00   98.40   98.39   98.80  100.80 
```

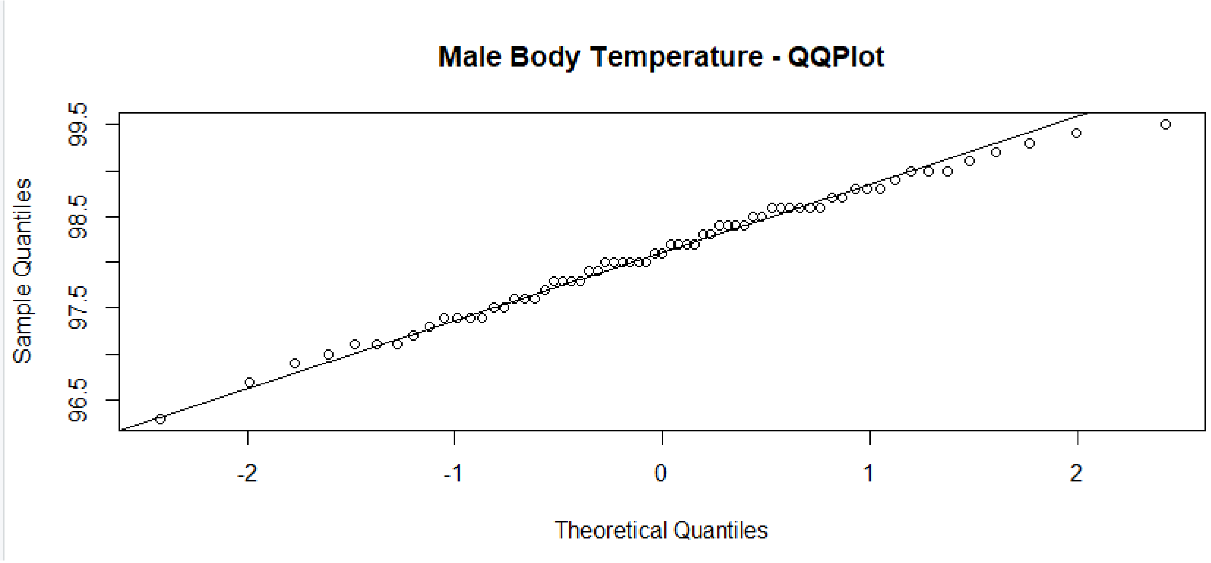
We can infer that the mean female body temperature is greater than mean male body temperature by 0.3. We then do a boxplot analysis.

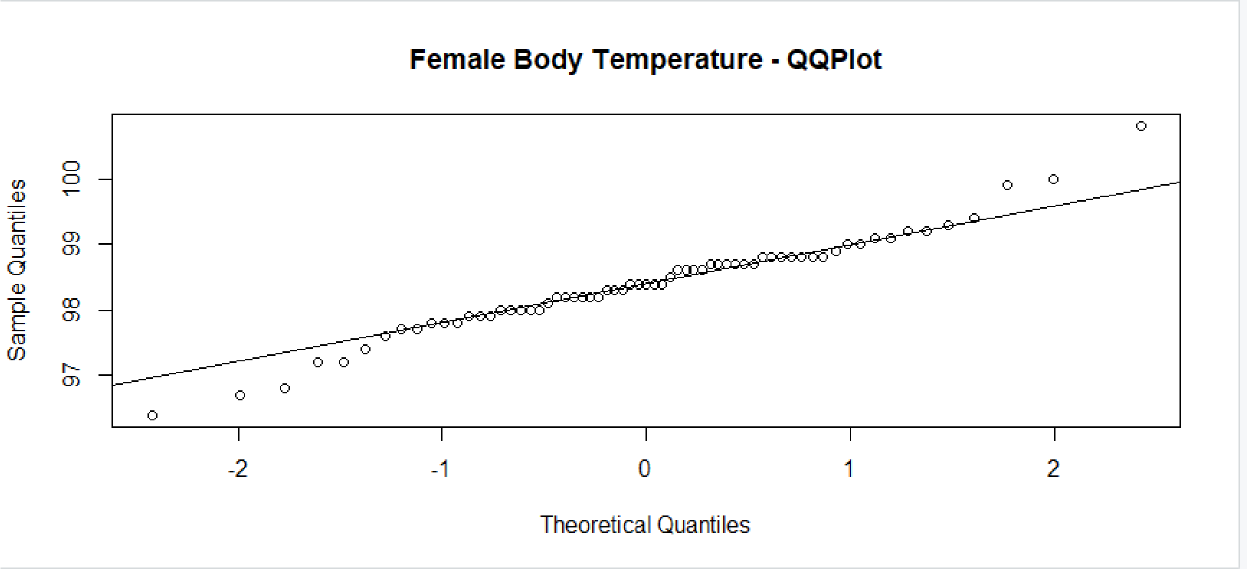


We can further infer that the there are four outliers in female body temperature whereas male has no outliers. We then perform a QQPlot analysis of male and female body temperature.

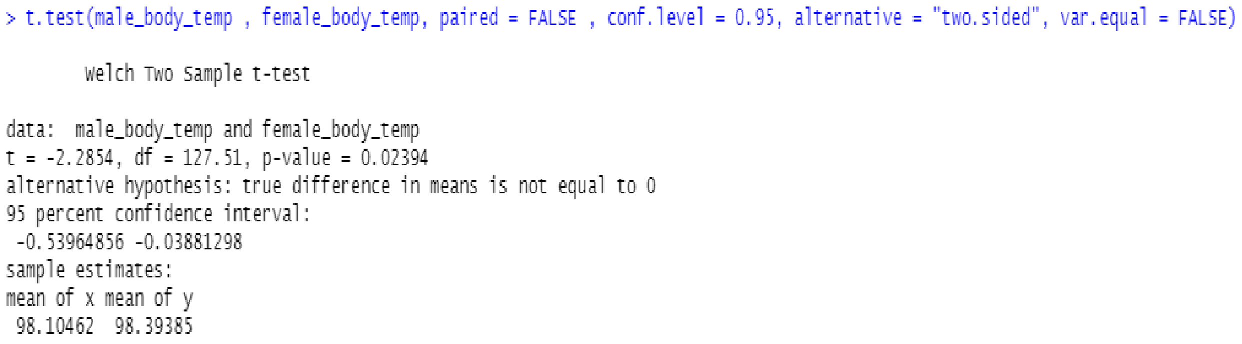


```
> boxplot(male_body_temp , female_body_temp , ylab = "Temperature", names = c("Male", "Female"))
> 
> qqnorm(male_body_temp , main = "Male Body Temperature - QQPlot")
> qqline(male_body_temp)
> 
> qqnorm(female_body_temp , main = "Female Body Temperature - QQPlot")
> qqline(female_body_temp)
```





From the QQplot, we assume normality based on the distribution of male and female body temperature. We then perform two sample T-Test for further analysis.

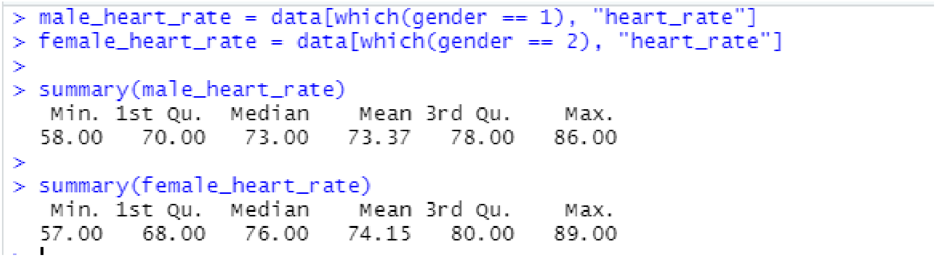


```
> t.test(male_body_temp , female_body_temp, paired = FALSE , conf.level = 0.95, alternative = "two.sided", var.equal = FALSE)

	Welch Two Sample t-test

data:  male_body_temp and female_body_temp
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 
```

Null Hypothesis : There is no difference in mean body temperature of male and female.
Alternative Hypothesis : There is a difference between mean body temperature of male and female.

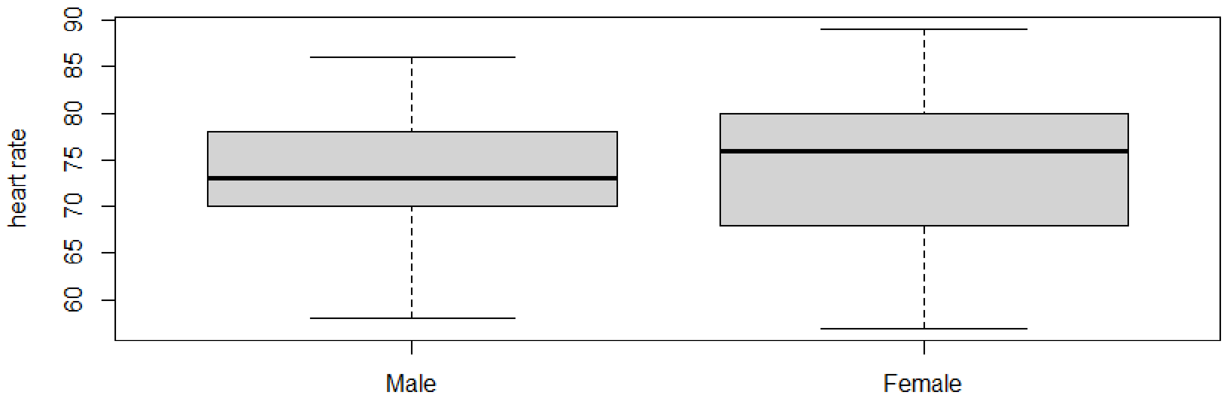
Since there is no enough evidence to support the null hypothesis, we can conclude that there is a difference between mean body temperature of male and female.

b) We filter the male and female heart rate based on ‘gender’ column from the given input data. Given below is the summary of male and female heart rates.

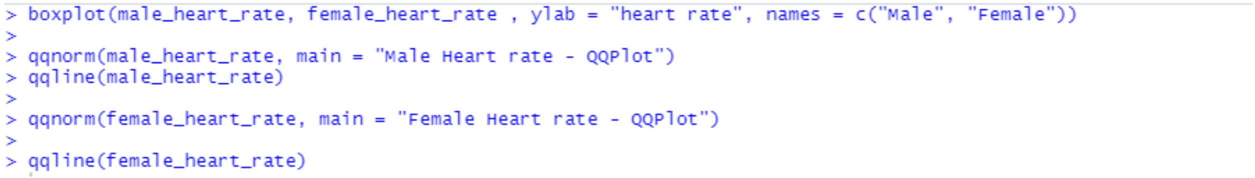


```
> male_heart_rate = data[which(gender == 1), "heart_rate"]
> female_heart_rate = data[which(gender == 2), "heart_rate"]
>
> summary(male_heart_rate)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  58.00   70.00   73.00   73.37   78.00   86.00 
>
> summary(female_heart_rate)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  57.00   68.00   76.00   74.15   80.00   89.00 
```

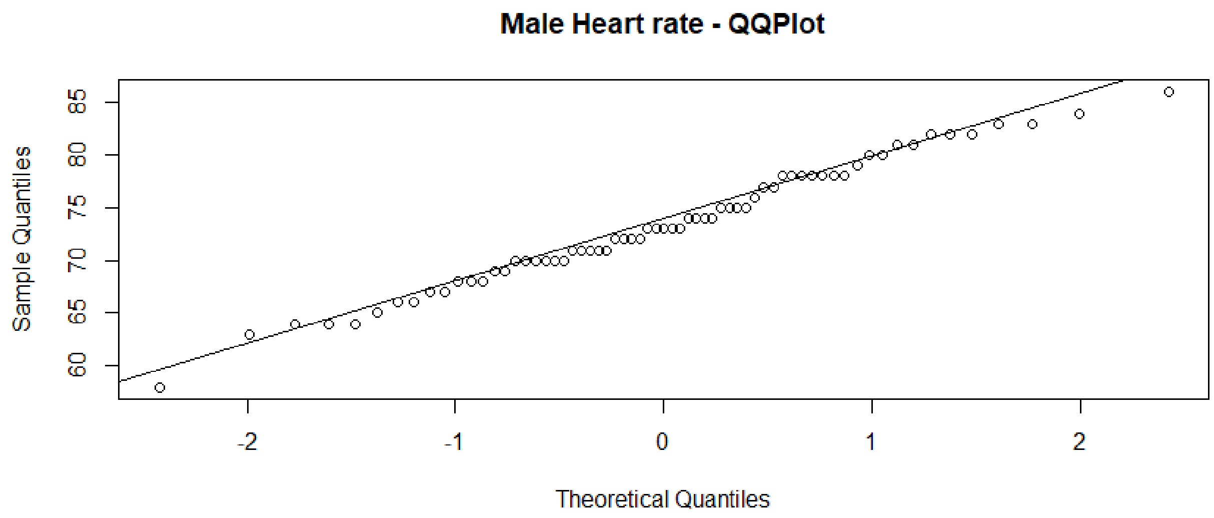
We can infer from the summary that the female heart rate is greater than male heart rate by 0.78. We then do a boxplot analysis.

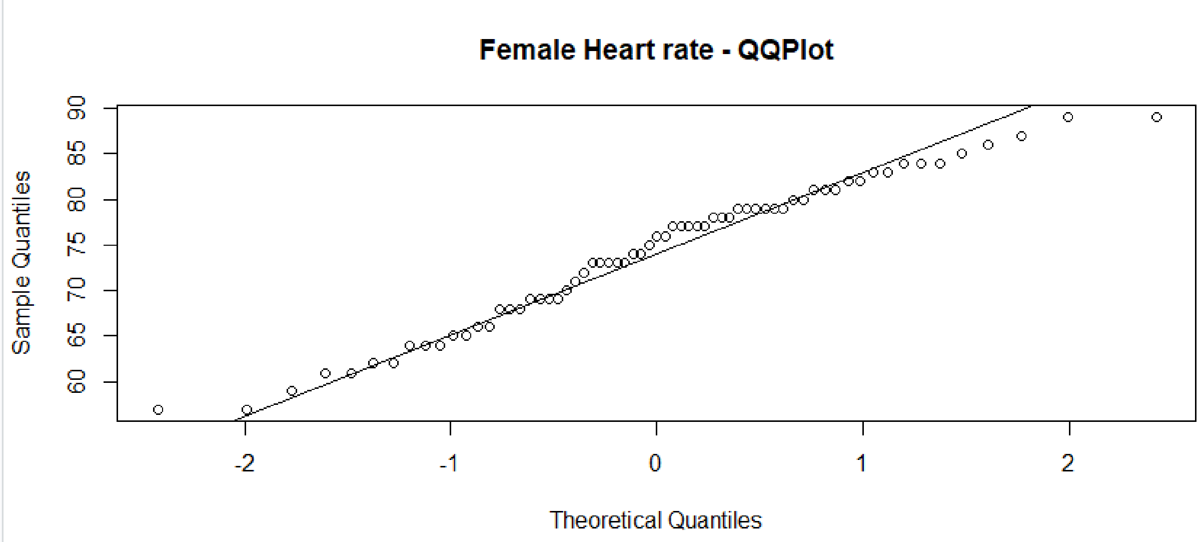


We can infer that the median of female heart rate is greater than that of male and there no outliers in both male and female heart rates. We then perform a QQPlot analysis of male and female heart rates.

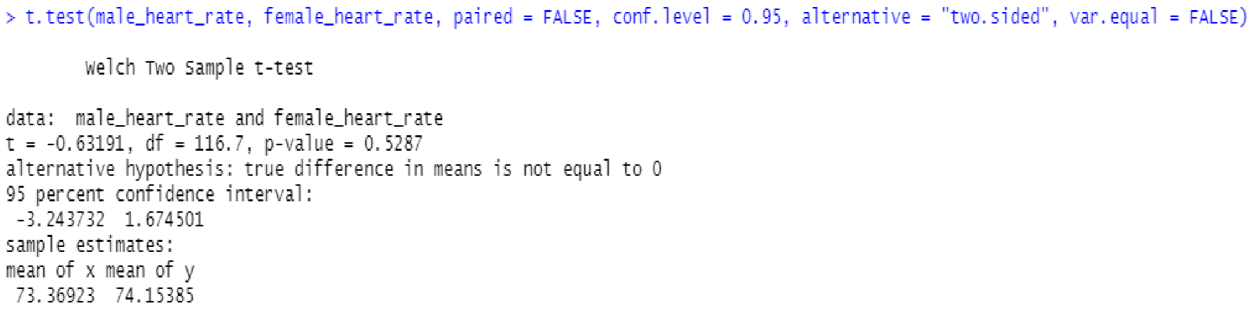


```
> boxplot(male_heart_rate, female_heart_rate , ylab = "heart rate", names = c("Male", "Female"))
>
> qqnorm(male_heart_rate, main = "Male Heart rate - QQPlot")
> qqline(male_heart_rate)
>
> qqnorm(female_heart_rate, main = "Female Heart rate - QQPlot")
>
> qqline(female_heart_rate)
```





From the QQplot, we assume normality based on the distribution of male and female heart rates. We then perform two sample T-Test for further analysis.

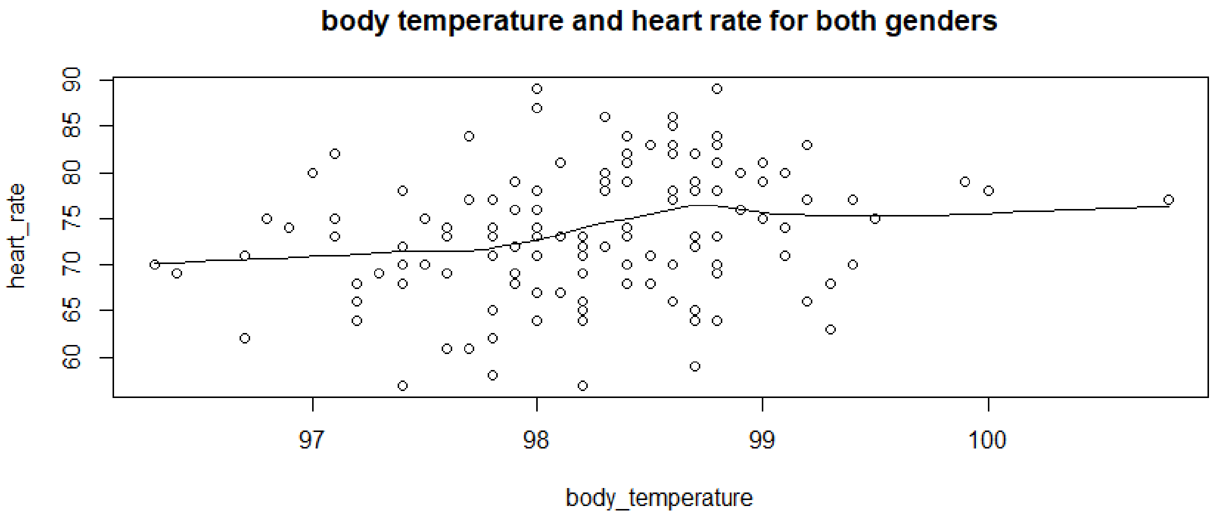


```
> t.test(male_heart_rate, female_heart_rate, paired = FALSE, conf.level = 0.95, alternative = "two.sided", var.equal = FALSE)

        Welch Two Sample t-test

data:  male_heart_rate and female_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 
```

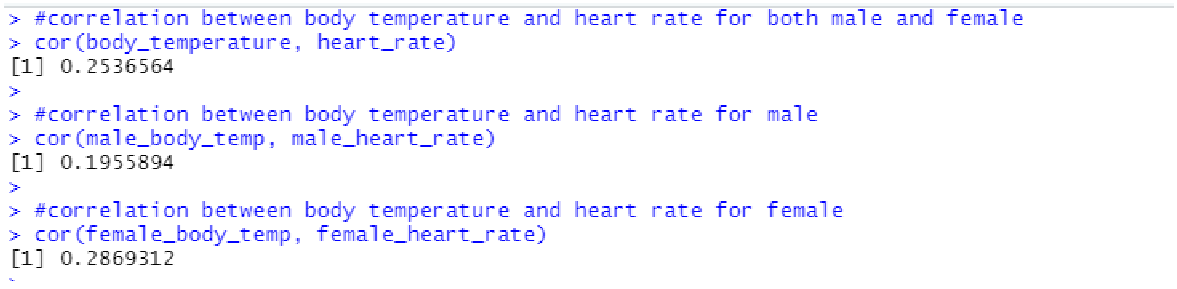
Null Hypothesis : There is no difference in mean heart rate of male and female.
Alternative Hypothesis : There is a difference between mean heart rate of male and female.

Since there is no enough evidence to support the null hypothesis, we can conclude that there is a difference between mean heart rate of male and female.

c) Initially, we do a scatter plot of body temperature and heart rate of both male and female.

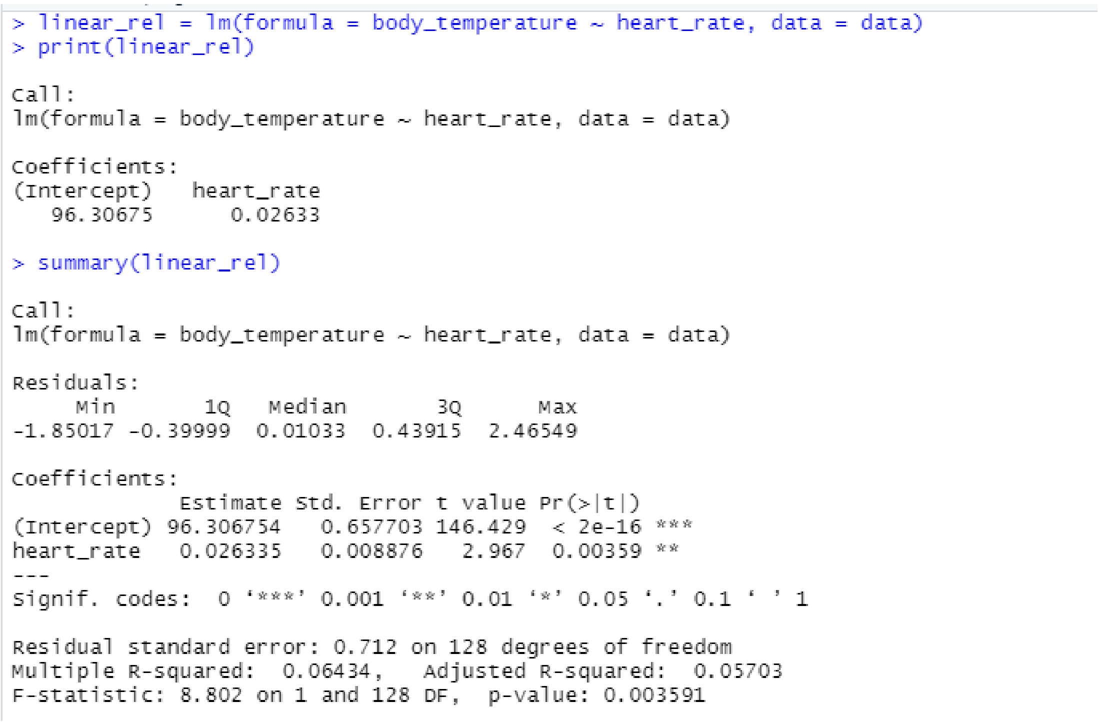


We can infer that there is a linear relationship between body tempearture and heart rate if we consider both female and male data. We also calculate the correlation of body temperature and heart rate of both male and female. We also calculate individual correlation of male and female data .

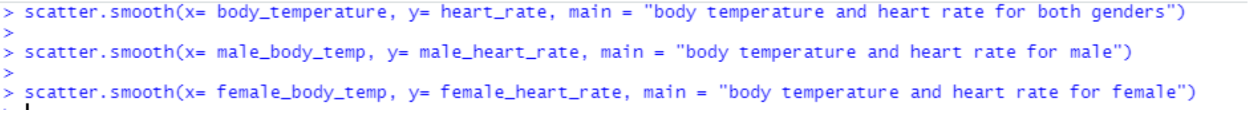


```
> #correlation between body temperature and heart rate for both male and female
> cor(body_temperature, heart_rate)
[1] 0.2536564
>
> #correlation between body temperature and heart rate for male
> cor(male_body_temp, male_heart_rate)
[1] 0.1955894
>
> #correlation between body temperature and heart rate for female
> cor(female_body_temp, female_heart_rate)
[1] 0.2869312
```

We then do a linear regression of data which includes both male and female body temperature and heart rate.

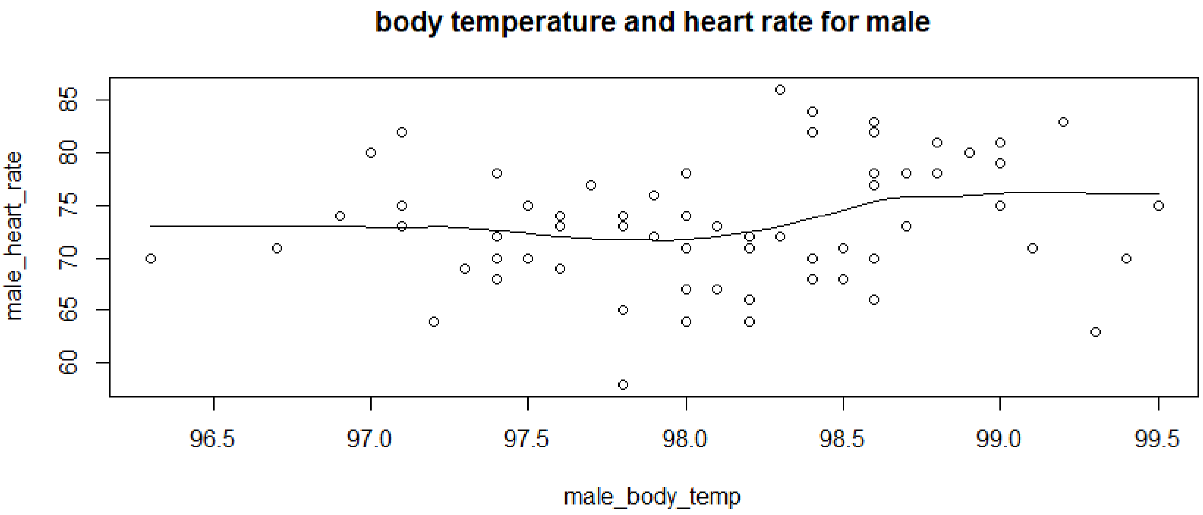


```
> linear_rel = lm(formula = body_temperature ~ heart_rate, data = data)
> print(linear_rel)

Call:
lm(formula = body_temperature ~ heart_rate, data = data)

Coefficients:
(Intercept)   heart_rate  
   96.30675      0.02633  

> summary(linear_rel)

Call:
lm(formula = body_temperature ~ heart_rate, data = data)

Residuals:
     Min       1Q   Median       3Q      Max 
-1.85017 -0.39999  0.01033  0.43915  2.46549 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 96.306754   0.657703 146.429  < 2e-16 ***
heart_rate   0.026335   0.008876   2.967  0.00359 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.712 on 128 degrees of freedom
Multiple R-squared:  0.06434,	Adjusted R-squared:  0.05703 
F-statistic: 8.802 on 1 and 128 DF,  p-value: 0.003591
```



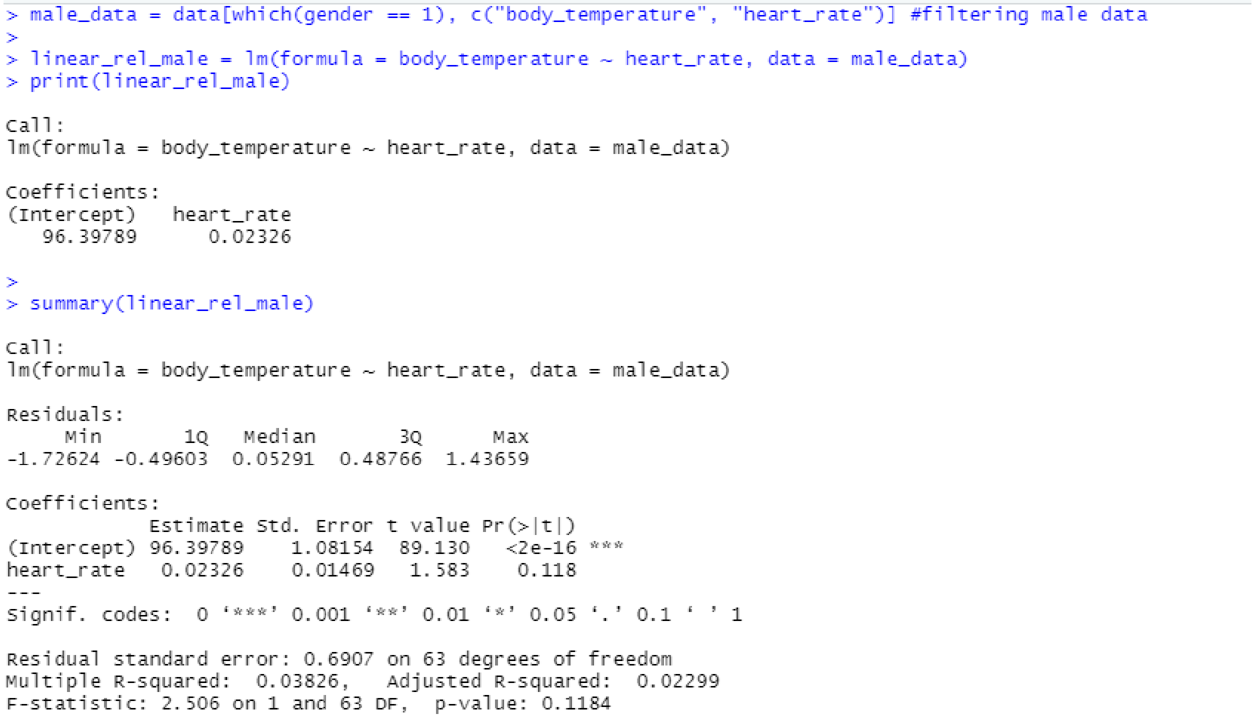
```
> scatter.smooth(x= body_temperature, y= heart_rate, main = "body temperature and heart rate for both genders")
> 
> scatter.smooth(x= male_body_temp, y= male_heart_rate, main = "body temperature and heart rate for male")
> 
> scatter.smooth(x= female_body_temp, y= female_heart_rate, main = "body temperature and heart rate for female")
```

We then do a scatterplot of body temperature and heart rate of only male data.



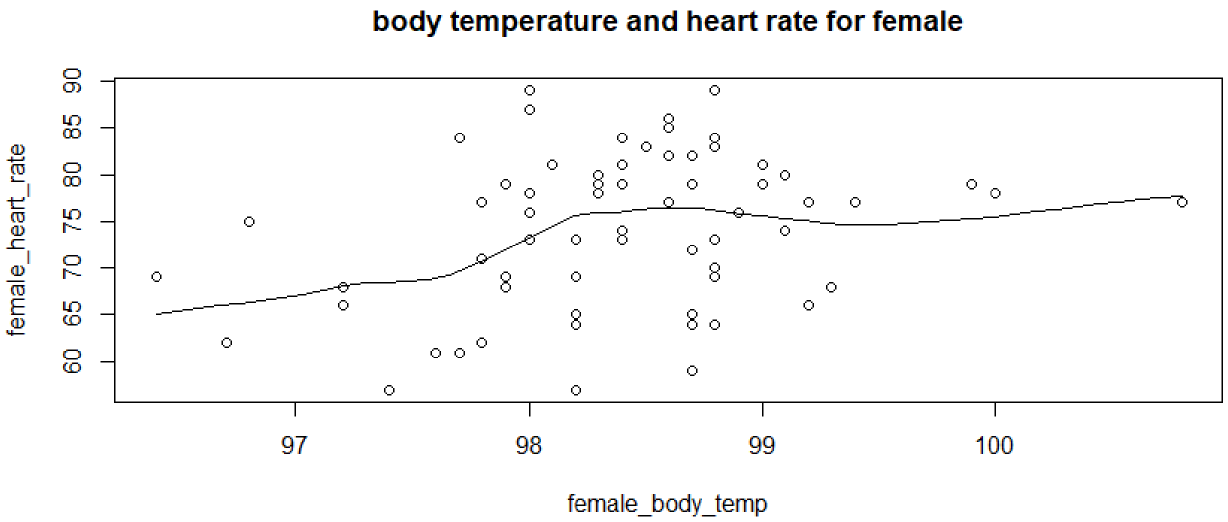
We can infer that there is a linear relationship between body temperature and heart rate if we consider only male data.

We then do a linear regression of these data for further analysis.



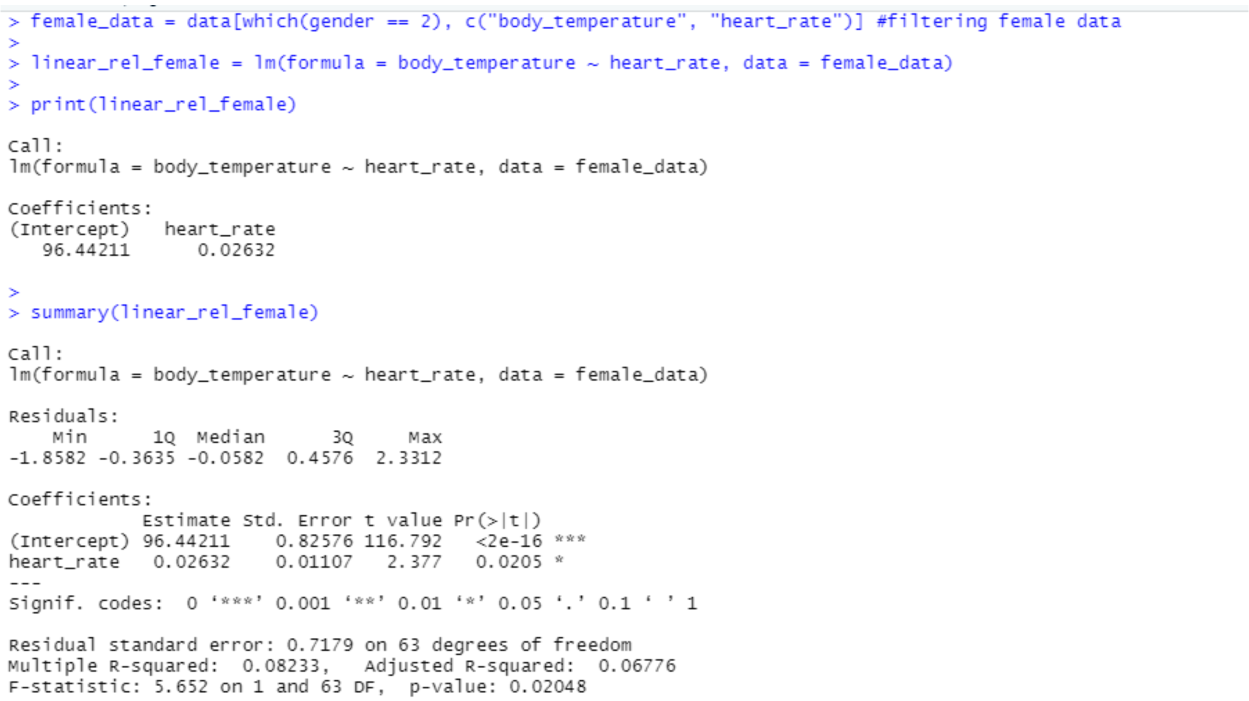
```
> male_data = data[which(gender == 1), c("body_temperature", "heart_rate")] #filtering male data
> 
> linear_rel_male = lm(formula = body_temperature ~ heart_rate, data = male_data)
> print(linear_rel_male)

Call:
lm(formula = body_temperature ~ heart_rate, data = male_data)

Coefficients:
(Intercept)   heart_rate  
   96.39789      0.02326  

> 
> summary(linear_rel_male)

Call:
lm(formula = body_temperature ~ heart_rate, data = male_data)

Residuals:
     Min       1Q   Median       3Q      Max 
-1.72624 -0.49603  0.05291  0.48766  1.43659 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 96.39789    1.08154  89.130   <2e-16 ***
heart_rate   0.02326    0.01469   1.583    0.118    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6907 on 63 degrees of freedom
Multiple R-squared:  0.03826,	Adjusted R-squared:  0.02299 
F-statistic: 2.506 on 1 and 63 DF,  p-value: 0.1184
```

We then do a scatterplot of body temperature and heart rate of only female data.



We can infer that there is a linear relationship between body temperature and heart rate if we consider only female data.

We then do a linear regression of these data for further analysis.



```
> female_data = data[which(gender == 2), c("body_temperature", "heart_rate")] #filtering female data
>
> linear_rel_female = lm(formula = body_temperature ~ heart_rate, data = female_data)
>
> print(linear_rel_female)

Call:
lm(formula = body_temperature ~ heart_rate, data = female_data)

Coefficients:
(Intercept)   heart_rate  
   96.44211      0.02632  

>
> summary(linear_rel_female)

Call:
lm(formula = body_temperature ~ heart_rate, data = female_data)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.8582 -0.3635 -0.0582  0.4576  2.3312 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 96.44211    0.82576 116.792   <2e-16 ***
heart_rate   0.02632    0.01107   2.377   0.0205 *  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7179 on 63 degrees of freedom
Multiple R-squared:  0.08233,	Adjusted R-squared:  0.06776 
F-statistic: 5.652 on 1 and 63 DF,  p-value: 0.02048
```

Finally, after scatterplot and linear regression analysis of body temperature and heart rate of both male and female data as well as analysing individual male and female data separately we can conclude that there is a a linear relationship between body temperature and heart rate. Since, the correlation between body temperature and heart rate of female is somewhat greater than that of male data, we can conclude that this relationship depends on gender.

2.

a) Using the functions given in Section 2, we have taken the value of n=10 and λ=0.1 to get the coverage probabilities as:

Z-Interval : 0.8778

Bootstrap Interval : 0.9226

b)

Repeating the above process for remaining values of n and λ, we get the following values:

Z – Proportion	L = 0.01	L = 0.1	L = 1	L = 10
N = 5	0.8164	0.8052	0.8088	0.8048
N = 10	0.8614	0.8778	0.8680	0.8672
N = 30	0.9174	0.9210	0.9220	0.9242
N = 100	0.9332	0.9384	0.9384	0.9376

B – Proportion	L = 0.01	L = 0.1	L = 1	L = 10
N = 5	0.8976	0.8974	0.8962	0.8982
N = 10	0.9132	0.9226	0.9174	0.9186
N = 30	0.9398	0.9324	0.9416	0.9358
N = 100	0.9478	0.9472	0.9486	0.9472

Graphically representation of the above data is given below,

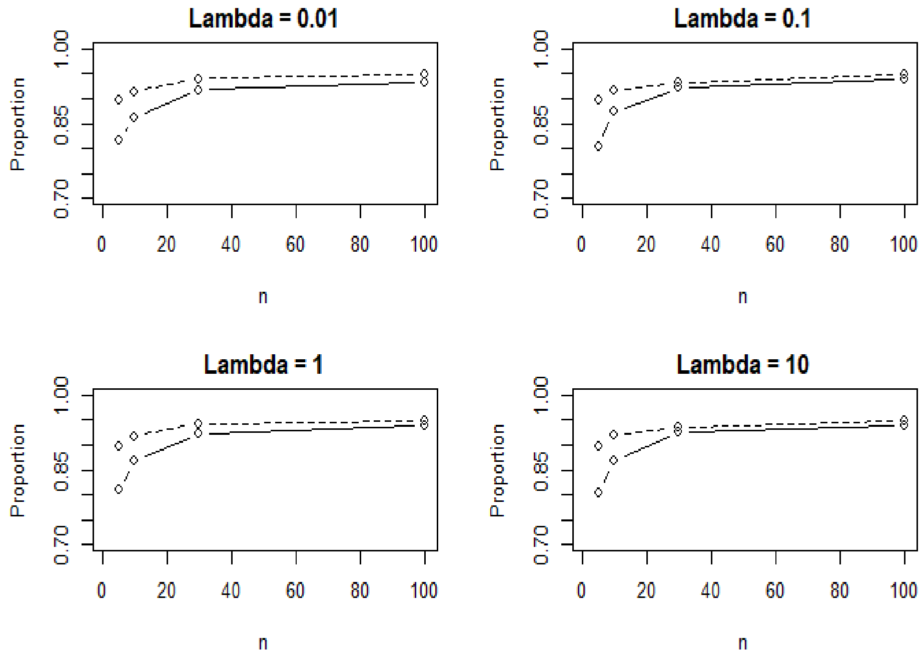


Figure 1 : The Solid line represents the Z-Proportion and the Dotted Line represents the Bootstrap Proportion. (Keeping λ fixed)

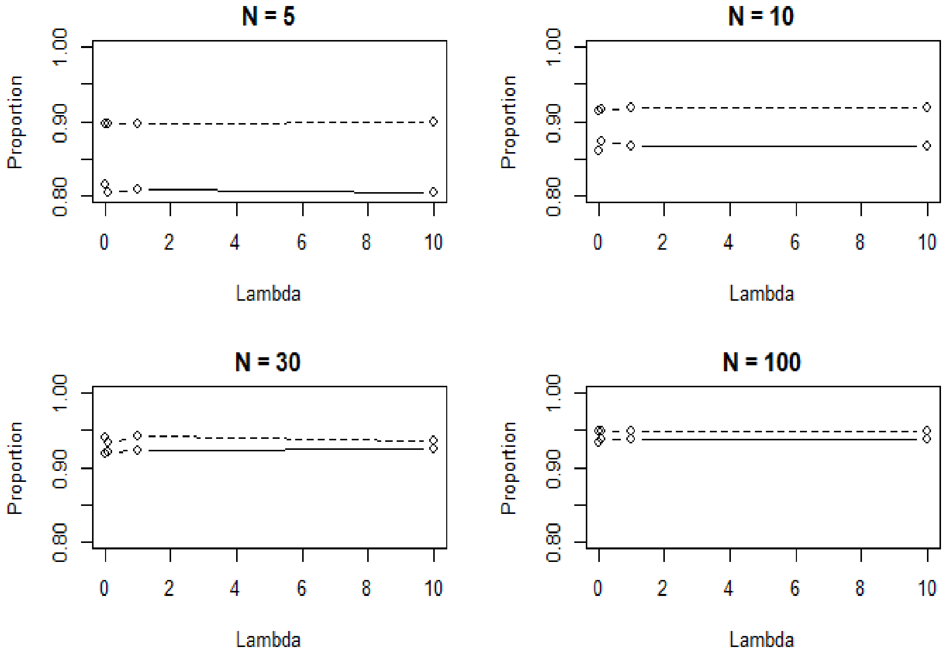


Figure 2 : The Solid line represents the Z-Proportion and the Dotted line represents the Bootstrap proportion. (Keeping N fixed)

Question 2 (c) :
From graphs in Figure 1, we can see that:
The graphs is similar when λ changes. Hence, we can say that the coverage probabilities do not depend on the value of λ.

The coverage probabilities that we get from z interval method are lower than those we get from the bootstrap method.

From graphs in Figure 2, we can see that:
The coverage probabilities depend on the value of n.

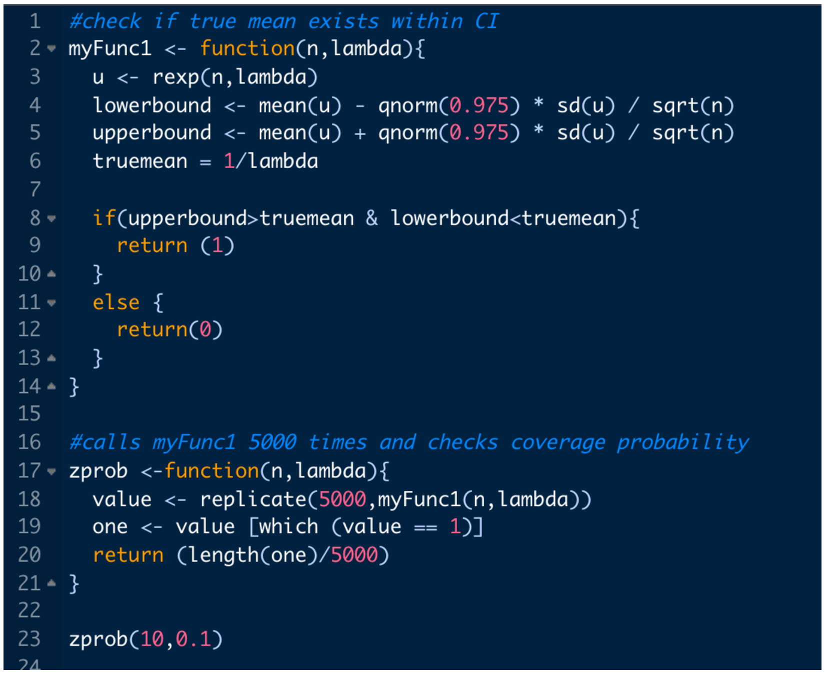
For large sample z interval, we could observe that the coverage probabilities are as accurate as the coverage probabilities that we obtained from the bootstrap method (when n is large)
N=30 onwards, the coverage probabilities for the bootstrap method are on the higher side.

Considering all the graphs, bootstrap method coverage probabilities are higher for any combination of n and λ than those of the large sample z interval method.
Thus, we could conclude that even for low values of n, the bootstrap method is more accurate.

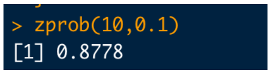
Question 2(d):

The conclusion in 2(c) does not depend upon values of λ because it is a parameter of the population distribution. Generally, conclusions about the convergence of sampling methods shouldn't depend on the population parameters. Also , by varying values of lambda (apart from given values), graph remains unchanged which implies that there is no dependency upon values of Lambda.

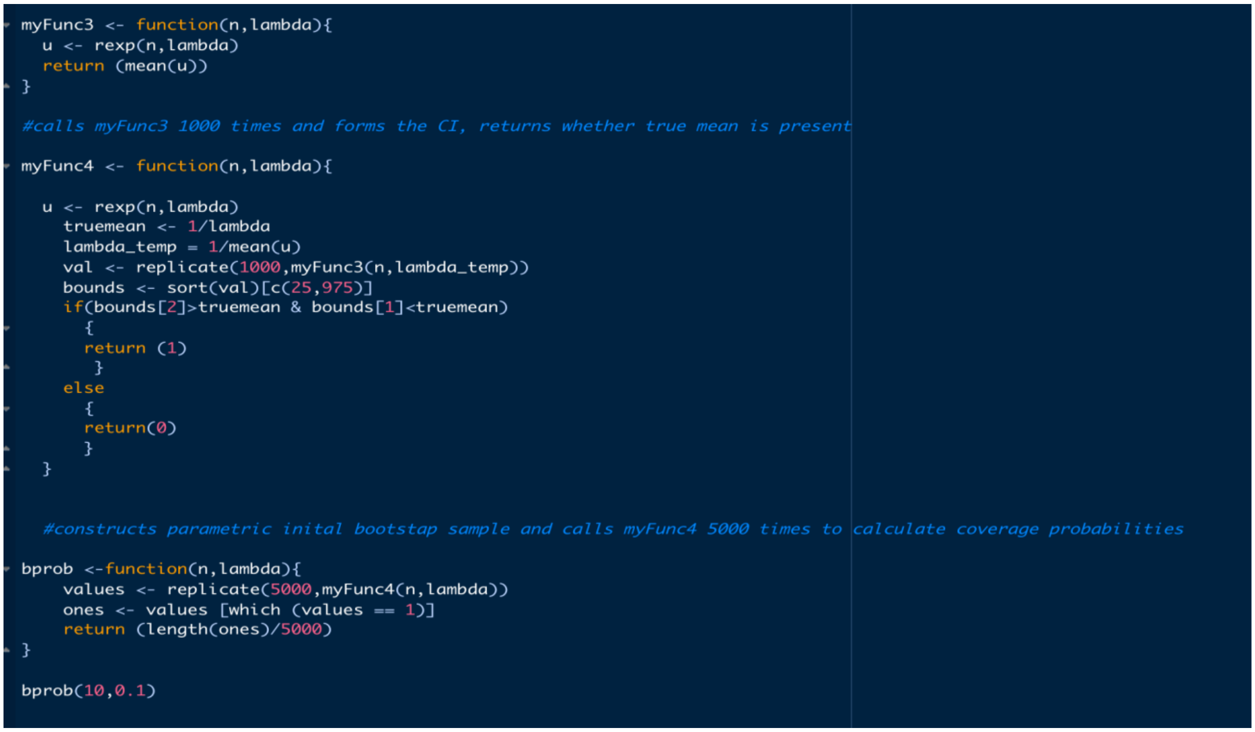
R- code :



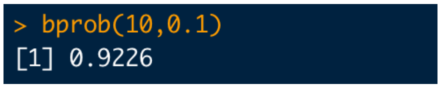
```r
#check if true mean exists within CI
myFunc1 <- function(n,lambda){
  u <- rexp(n,lambda)
  lowerbound <- mean(u) - qnorm(0.975) * sd(u) / sqrt(n)
  upperbound <- mean(u) + qnorm(0.975) * sd(u) / sqrt(n)
  truemean = 1/lambda

  if(upperbound>truemean & lowerbound<truemean){
    return (1)
  }
  else {
    return(0)
  }
}

#calls myFunc1 5000 times and checks coverage probability
zprob <-function(n,lambda){
  value <- replicate(5000,myFunc1(n,lambda))
  one <- value [which (value == 1)]
  return (length(one)/5000)
}

zprob(10,0.1)
```



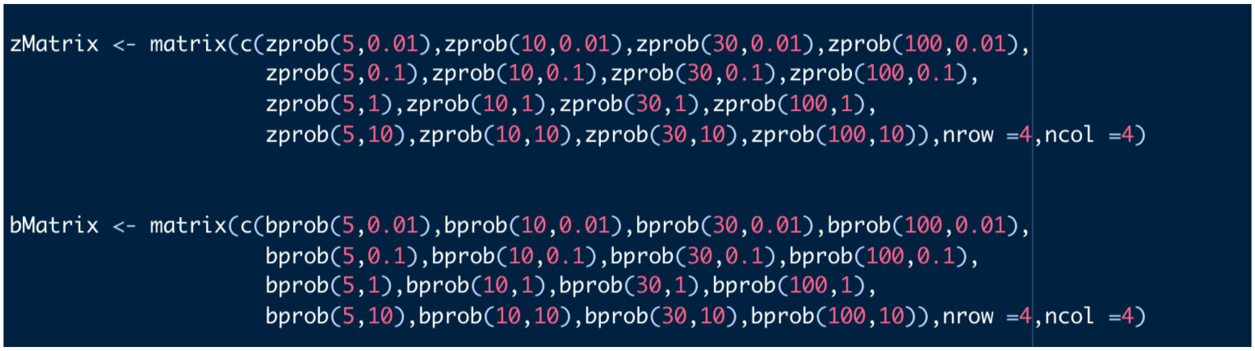
```
> zprob(10,0.1)
[1] 0.8778
```



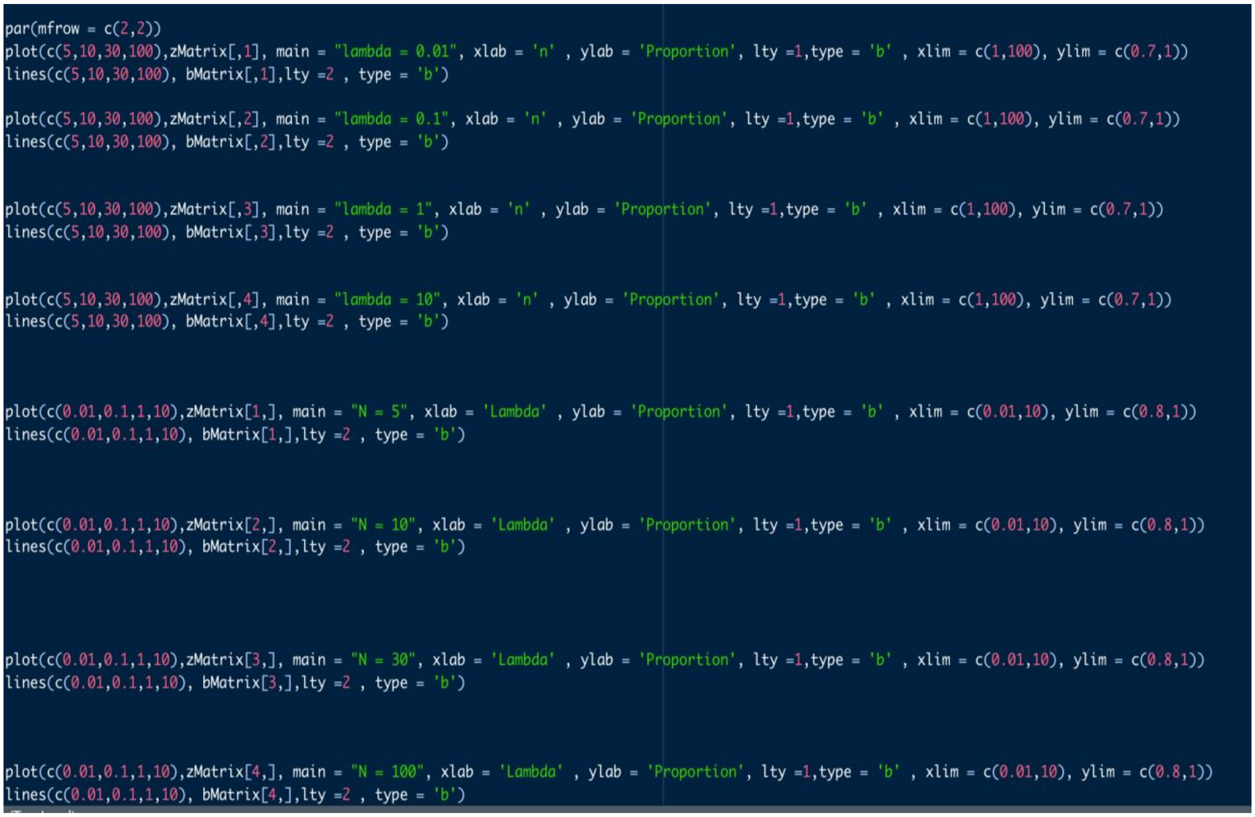
```
myFunc3 <- function(n,lambda){
  u <- rexp(n,lambda)
  return (mean(u))
}

#calls myFunc3 1000 times and forms the CI, returns whether true mean is present

myFunc4 <- function(n,lambda){

  u <- rexp(n,lambda)
    truemean <- 1/lambda
    lambda_temp = 1/mean(u)
    val <- replicate(1000,myFunc3(n,lambda_temp))
    bounds <- sort(val)[c(25,975)]
    if(bounds[2]>truemean & bounds[1]<truemean)
      {
      return (1)
       }
    else
      {
      return(0)
      }
  }


  #constructs parametric inital bootstap sample and calls myFunc4 5000 times to calculate coverage probabilities

bprob <-function(n,lambda){
    values <- replicate(5000,myFunc4(n,lambda))
    ones <- values [which (values == 1)]
    return (length(ones)/5000)
}

bprob(10,0.1)
```



```
> bprob(10,0.1)
[1] 0.9226
```



```
zMatrix <- matrix(c(zprob(5,0.01),zprob(10,0.01),zprob(30,0.01),zprob(100,0.01),
                    zprob(5,0.1),zprob(10,0.1),zprob(30,0.1),zprob(100,0.1),
                    zprob(5,1),zprob(10,1),zprob(30,1),zprob(100,1),
                    zprob(5,10),zprob(10,10),zprob(30,10),zprob(100,10)),nrow =4,ncol =4)


bMatrix <- matrix(c(bprob(5,0.01),bprob(10,0.01),bprob(30,0.01),bprob(100,0.01),
                    bprob(5,0.1),bprob(10,0.1),bprob(30,0.1),bprob(100,0.1),
                    bprob(5,1),bprob(10,1),bprob(30,1),bprob(100,1),
                    bprob(5,10),bprob(10,10),bprob(30,10),bprob(100,10)),nrow =4,ncol =4)
```



```r
par(mfrow = c(2,2))
plot(c(5,10,30,100),zMatrix[,1], main = "lambda = 0.01", xlab = 'n' , ylab = 'Proportion', lty =1,type = 'b' , xlim = c(1,100), ylim = c(0.7,1))
lines(c(5,10,30,100), bMatrix[,1],lty =2 , type = 'b')

plot(c(5,10,30,100),zMatrix[,2], main = "lambda = 0.1", xlab = 'n' , ylab = 'Proportion', lty =1,type = 'b' , xlim = c(1,100), ylim = c(0.7,1))
lines(c(5,10,30,100), bMatrix[,2],lty =2 , type = 'b')


plot(c(5,10,30,100),zMatrix[,3], main = "lambda = 1", xlab = 'n' , ylab = 'Proportion', lty =1,type = 'b' , xlim = c(1,100), ylim = c(0.7,1))
lines(c(5,10,30,100), bMatrix[,3],lty =2 , type = 'b')


plot(c(5,10,30,100),zMatrix[,4], main = "lambda = 10", xlab = 'n' , ylab = 'Proportion', lty =1,type = 'b' , xlim = c(1,100), ylim = c(0.7,1))
lines(c(5,10,30,100), bMatrix[,4],lty =2 , type = 'b')



plot(c(0.01,0.1,1,10),zMatrix[1,], main = "N = 5", xlab = 'Lambda' , ylab = 'Proportion', lty =1,type = 'b' , xlim = c(0.01,10), ylim = c(0.8,1))
lines(c(0.01,0.1,1,10), bMatrix[1,],lty =2 , type = 'b')



plot(c(0.01,0.1,1,10),zMatrix[2,], main = "N = 10", xlab = 'Lambda' , ylab = 'Proportion', lty =1,type = 'b' , xlim = c(0.01,10), ylim = c(0.8,1))
lines(c(0.01,0.1,1,10), bMatrix[2,],lty =2 , type = 'b')




plot(c(0.01,0.1,1,10),zMatrix[3,], main = "N = 30", xlab = 'Lambda' , ylab = 'Proportion', lty =1,type = 'b' , xlim = c(0.01,10), ylim = c(0.8,1))
lines(c(0.01,0.1,1,10), bMatrix[3,],lty =2 , type = 'b')



plot(c(0.01,0.1,1,10),zMatrix[4,], main = "N = 100", xlab = 'Lambda' , ylab = 'Proportion', lty =1,type = 'b' , xlim = c(0.01,10), ylim = c(0.8,1))
lines(c(0.01,0.1,1,10), bMatrix[4,],lty =2 , type = 'b')
```