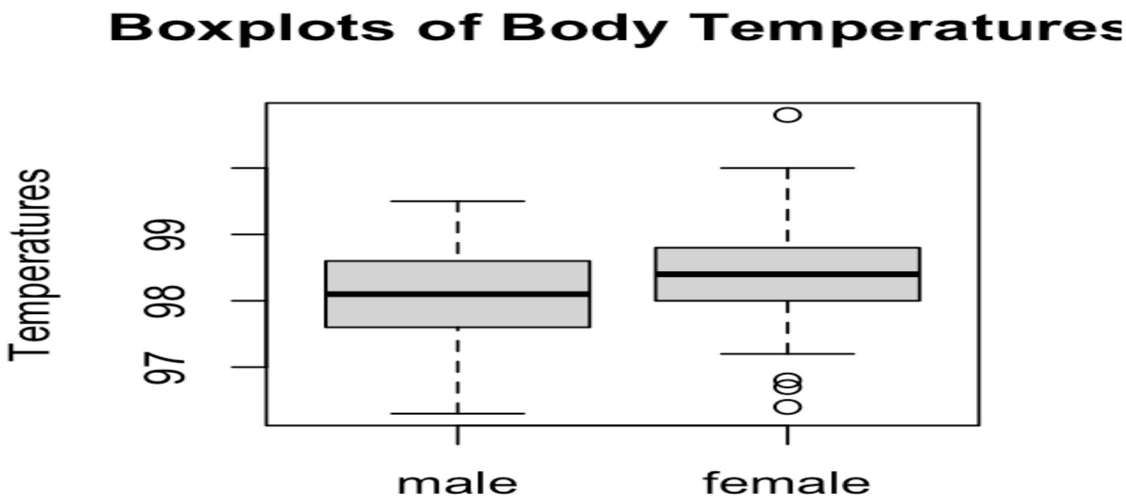
Mini Project 5

Group Members :: Kavya Chowdhary Cherukuri and Kaushik Nadimpalli

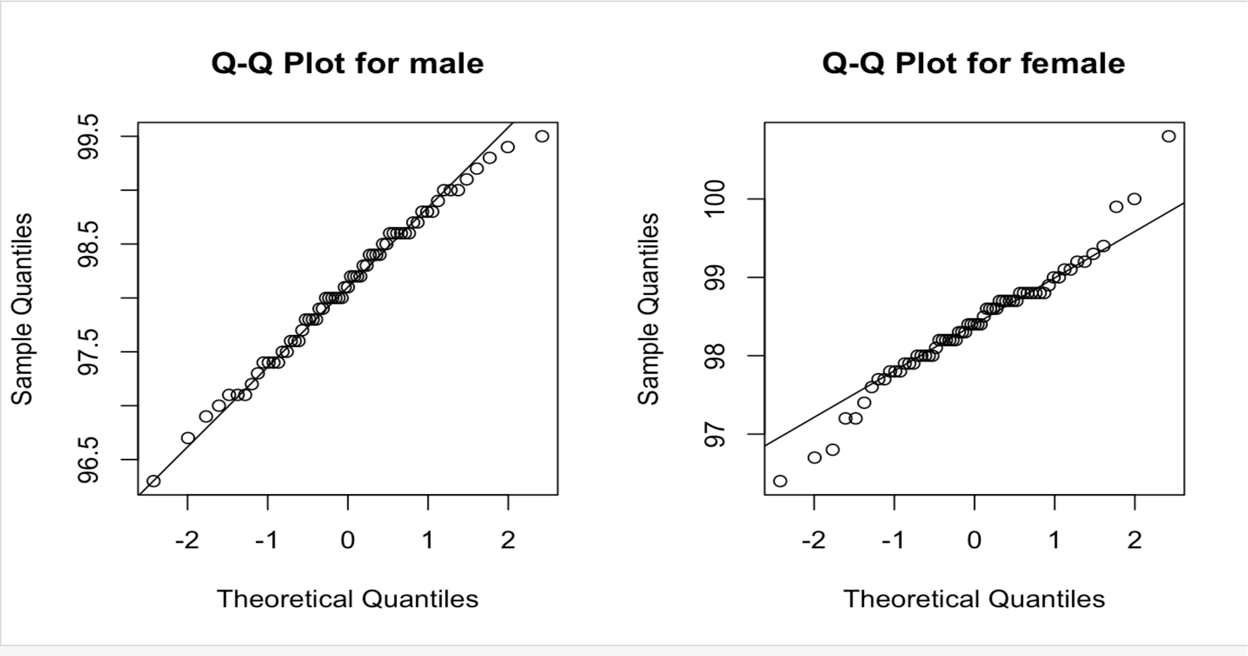
Contribution :: Both team members worked on the project. Kavya worked on Question 1 and Kaushik worked on Question 2. When it came to the report, both discussed the results from the code and developed the report for their respective question portions.

Section 1 - Solutions

Question 1a



From the Box plot, we can see that there are more outliers for females when compared to male and the medians Q1 and Q2 are greater for females. From this we can say that females have a little higher mean than male.



From the Q-Q plots, we can see that for the body temperature, the distribution is normal.

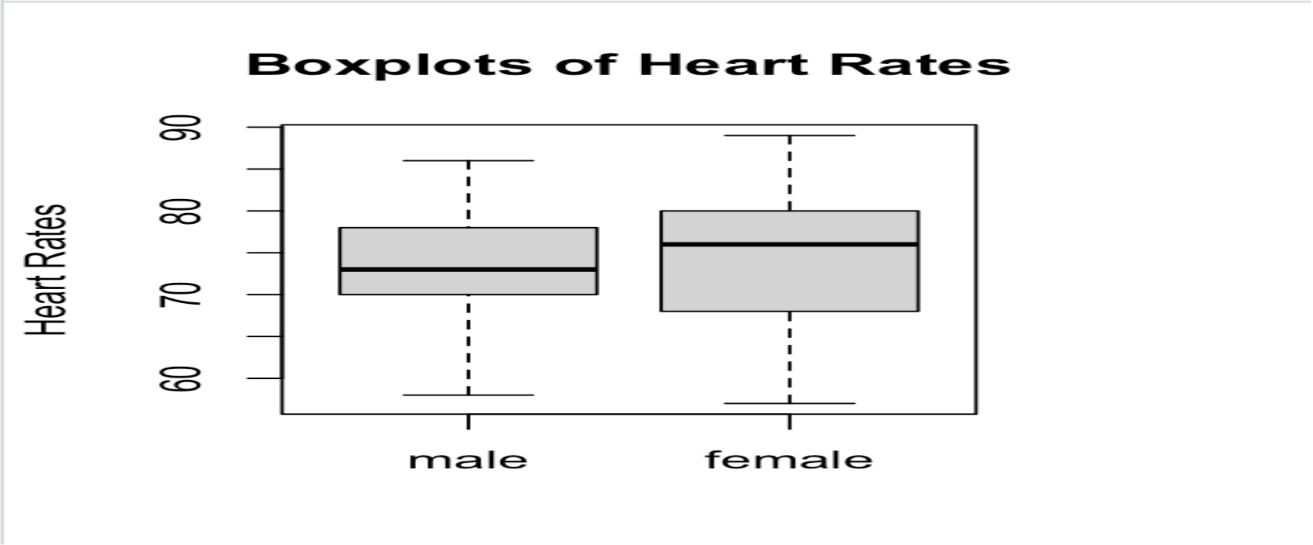
Null Hypothesis: the difference between both the means is 0
Alternate Hypothesis: the difference between both the means is not equals to 0.

The samples here are used as independent with irregular variances, hence t-distribution can be used. We build the confidence interval by using the t.test function in R.

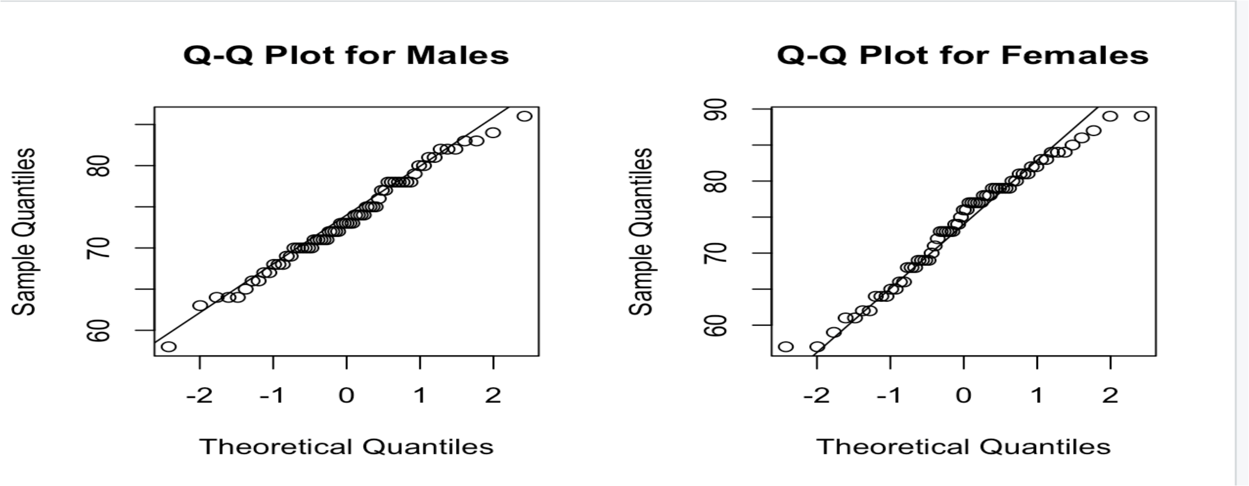
We see that the confidence interval of the 't.test' function is:
(-0.53964856, -0.03881298)

The p-value is **0.02394**
As the p-value is smaller than 0.05 and 0 is not in the confidence interval we **reject** the null hypothesis. Therefore, we can say that the mean of female body temperature is slightly greater than male body temperature.

Question 1b



Here from the Box plot, we see that the Median and the Q3 are greater for Females, whereas Q1 is greater for Males.



From the depicted Q-Q plots, it is clear that the distribution is nearly normal for both the values for the heart rate.

If 'M' and 'F' are the heart rate of males and females respectively. Then the same mean estimate of the male population and the female population is m and f respectively.

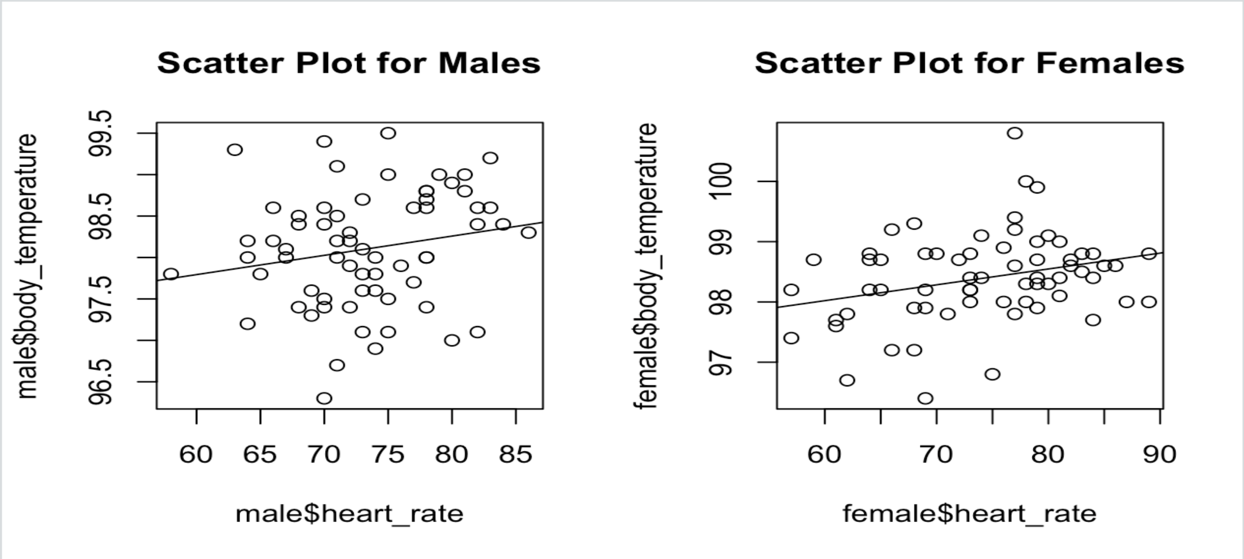
Null Hypothesis: the difference between both the means is 0
Alternate Hypothesis: the difference between both the means is not equals to 0.

The samples here are used as independent with irregular variances, hence t-distribution can be used. We build the confidence interval by using the t.test function in R.

We see that the confidence interval of the 't.test' function is:
(-3.243732, 1.674501)
The p-value we got is **0.5287.**

As the p-value is greater than 0.05 and 0 is in the confidence interval, we **accept** the null hypothesis. Therefore, we say that the heart rate means females and males are equal.

Question 1c



We can notice that the line of fit in the plots above is greater than 0. Hence, there is a positive association between body temp and heart rates.

The relationship between the heart rate and the body temperature can be determined by using the correlation function 'cor(X, Y)'.

The correlation between heart rate and body temperature is as follows:
For males: **0.1955894**
For females: **0.2869312**

Here we can say that the correlation is more effective when the value is higher, while also concluding that the relation between the body temperature and the heart rate is weak. Therefore, we can conclude that the correlation for body temperature and heart rate for females is a little **stronger** than that correlation for males.

Question 2a

We are considering 2 methods here for the CI - large sample Z-interval (interval 1) and the parametric bootstrap method (interval 2). In our code, we have done just that with the large sample CI construction and repetition 5000 times first, and then the same but with the bootstrap method.

Please refer to code for more details and annotations on how this was done.

Our combination is $(n,\lambda) = (5, 0.01)$, for which we get the following coverage probabilities ::

Z-interval: 0.8056
Bootstrap Interval: 0.8960

Question 2b

Below, we have showcased the results of the remaining combinations for (n,λ) in a tabular format:

Interval 1 Iterations

Large-Scale Z-proportions	$\lambda = 0.01$	$\lambda = 0.01$	$\lambda = 0.01$	$\lambda = 0.01$
n = 5	.8072	.8158	.8144	.8120
n = 10	.8816	.8788	.8754	,8726
n = 30	.9164	.9162	.9144	.9210
n = 100	.9410	.9412	.9390	.9344

Interval 2 Iterations

Bootstrap proportions	$\lambda = 0.01$	$\lambda = 0.01$	$\lambda = 0.01$	$\lambda = 0.01$
n = 5	.8998	.8912	0.9008	.8926
n = 10	0.9248	0.9208	0.9202	0.9210
n = 30	0.9406	0.9456	0.9352	0.9338
n = 100	0.9430	0.9466	0.9444	0.9468

Question 2c

For large scale Z-interval, n = 100 or above seems to be needed for the interval to be accurate. Likewise, for bootstrap intervals, n=30 or greater seem to give a good accurate intervals.

The coverage probabilities do not necessarily depend on λ as we can see in both the intervals. We notice that there seems to not be a specific linear relationship between the coverage probability with λ.

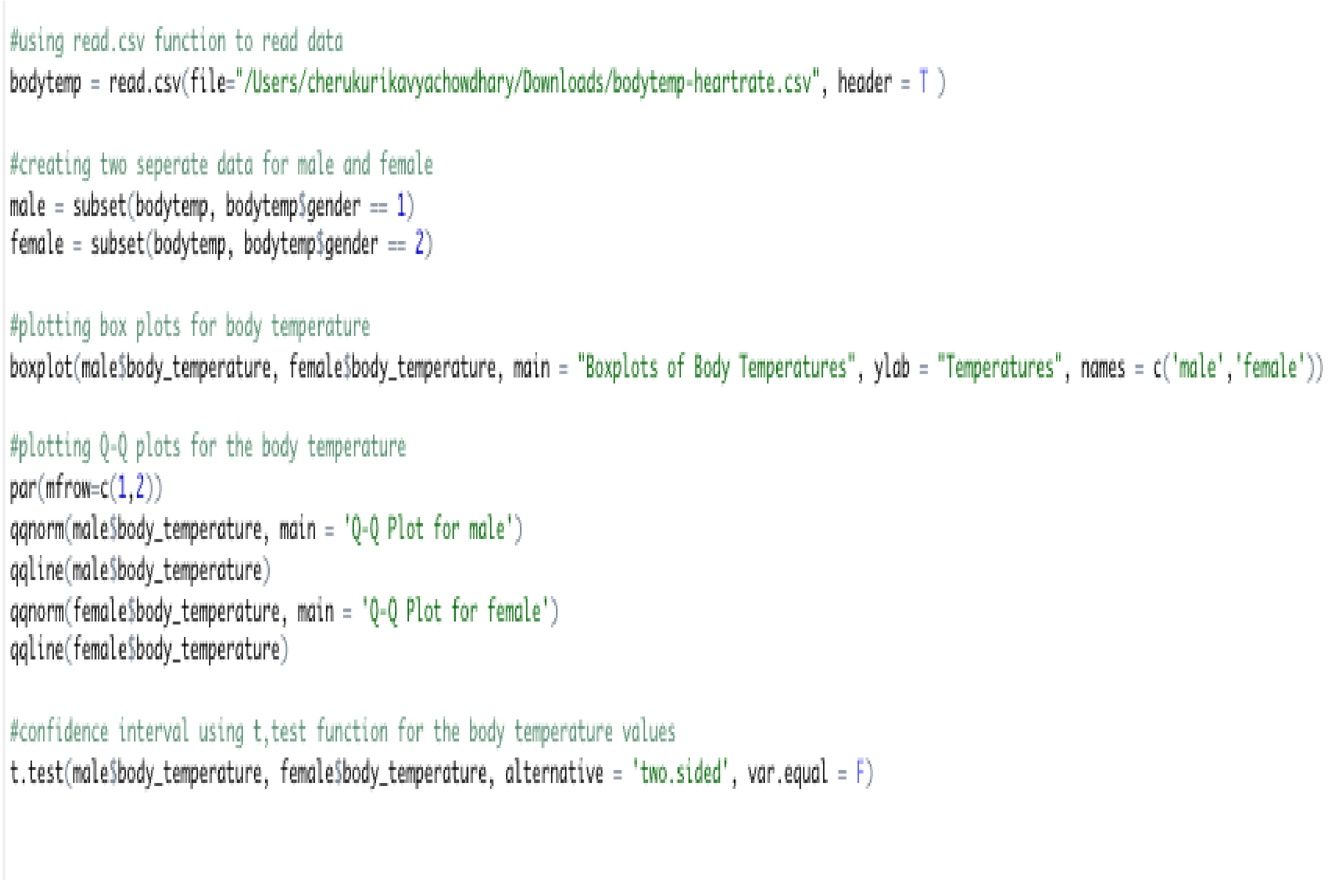
Overall, we can see that there are higher coverage probabilities with the bootstrap method compared to the large-scale z-interval method - even for small values of n. Hence we would recommend the bootstrap method.

Question 2d

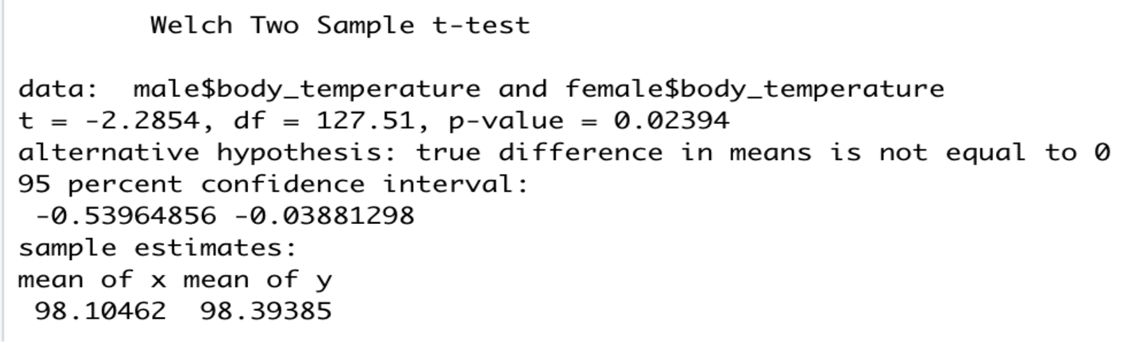
From our analysis, we do not necessarily think the conclusions are based on specific values of fixed λ, at-least not all of them. This is because we are getting different coverage probabilities when we run the values again with same (n, λ) combination. When we re-ran the iterations again, we tried to verify this and it does not always hold for specific values of lambda.

Section 2 - R Code and Annotations

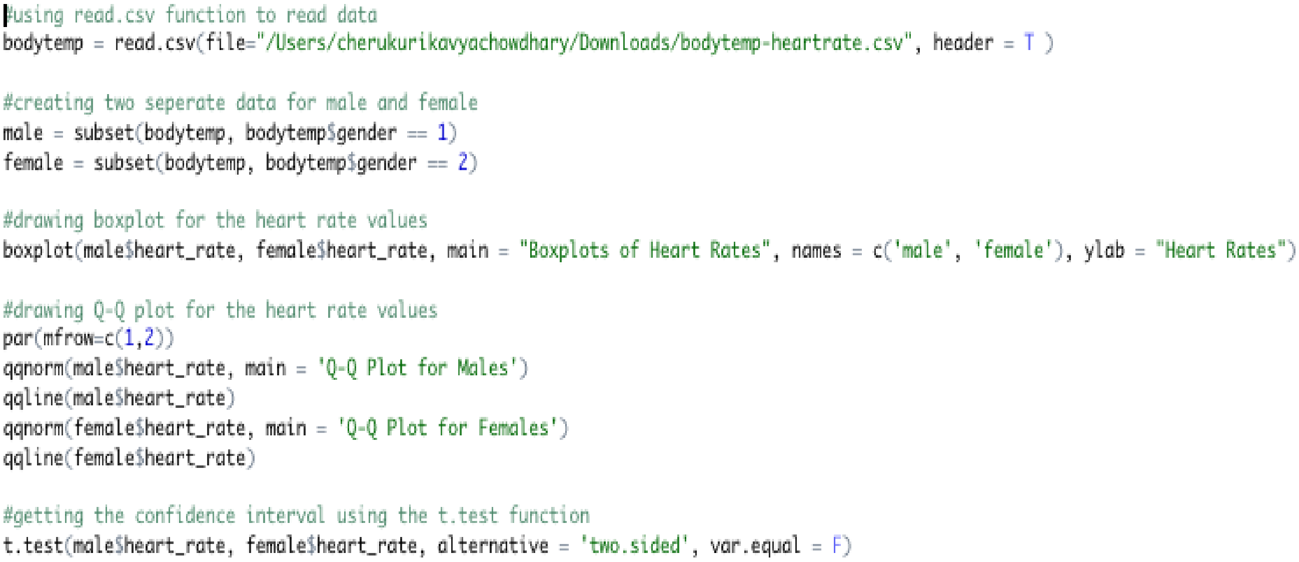
Question 1 Code and Comments



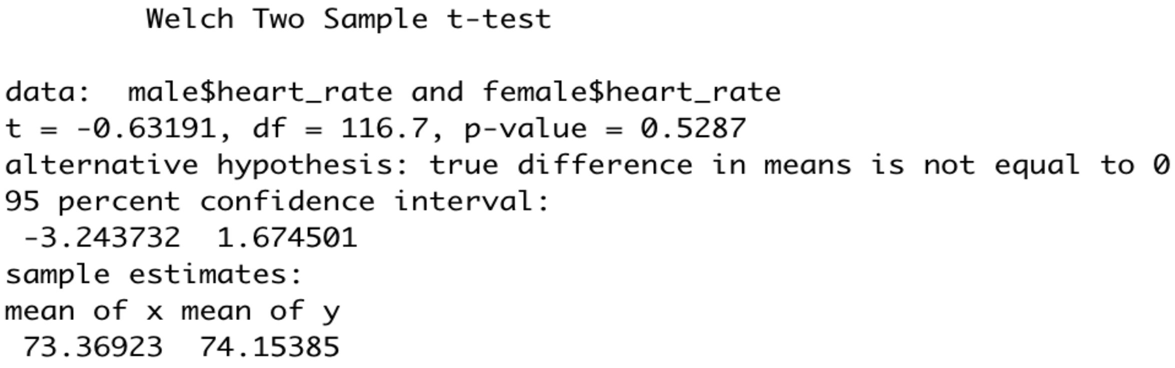
```r
#using read.csv function to read data
bodytemp = read.csv(file="/Users/cherukurikavyachowdhary/Downloads/bodytemp-heartrate.csv", header = T )

#creating two seperate data for male and female
male = subset(bodytemp, bodytemp$gender == 1)
female = subset(bodytemp, bodytemp$gender == 2)

#plotting box plots for body temperature
boxplot(male$body_temperature, female$body_temperature, main = "Boxplots of Body Temperatures", ylab = "Temperatures", names = c('male','female'))

#plotting Q-Q plots for the body temperature
par(mfrow=c(1,2))
qqnorm(male$body_temperature, main = 'Q-Q Plot for male')
qqline(male$body_temperature)
qqnorm(female$body_temperature, main = 'Q-Q Plot for female')
qqline(female$body_temperature)

#confidence interval using t,test function for the body temperature values
t.test(male$body_temperature, female$body_temperature, alternative = 'two.sided', var.equal = F)
```



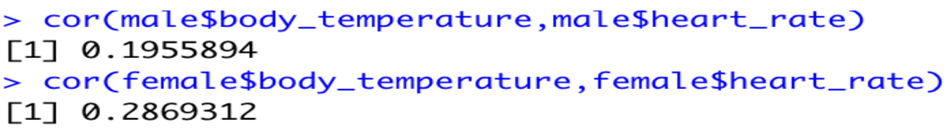
```
        Welch Two Sample t-test

data:  male$body_temperature and female$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 
```



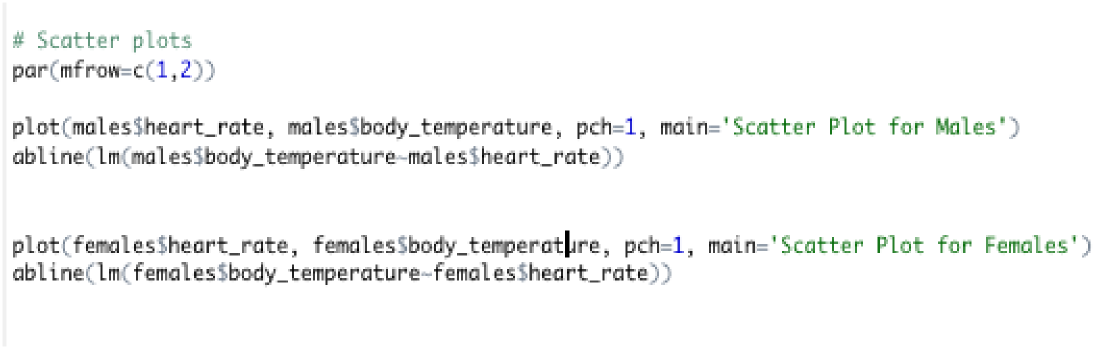
```
#using read.csv function to read data
bodytemp = read.csv(file="/Users/cherukurikavyachowdhary/Downloads/bodytemp-heartrate.csv", header = T )

#creating two seperate data for male and female
male = subset(bodytemp, bodytemp$gender == 1)
female = subset(bodytemp, bodytemp$gender == 2)

#drawing boxplot for the heart rate values
boxplot(male$heart_rate, female$heart_rate, main = "Boxplots of Heart Rates", names = c('male', 'female'), ylab = "Heart Rates")

#drawing Q-Q plot for the heart rate values
par(mfrow=c(1,2))
qqnorm(male$heart_rate, main = 'Q-Q Plot for Males')
qqline(male$heart_rate)
qqnorm(female$heart_rate, main = 'Q-Q Plot for Females')
qqline(female$heart_rate)

#getting the confidence interval using the t.test function
t.test(male$heart_rate, female$heart_rate, alternative = 'two.sided', var.equal = F)
```



```
        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 
```

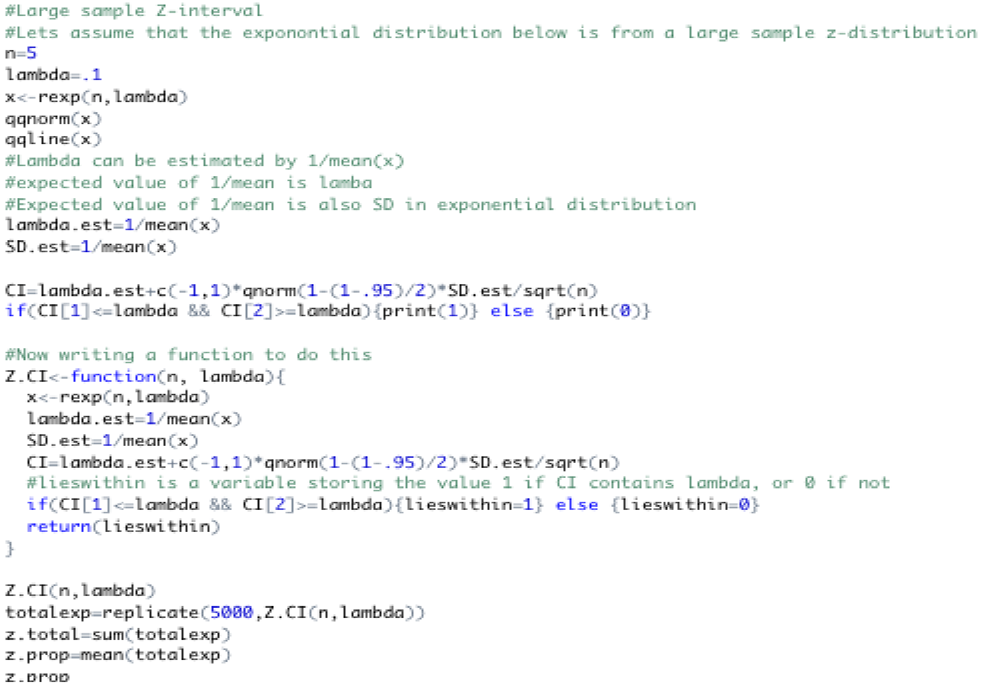


```
> cor(male$body_temperature,male$heart_rate)
[1] 0.1955894
> cor(female$body_temperature,female$heart_rate)
[1] 0.2869312
```

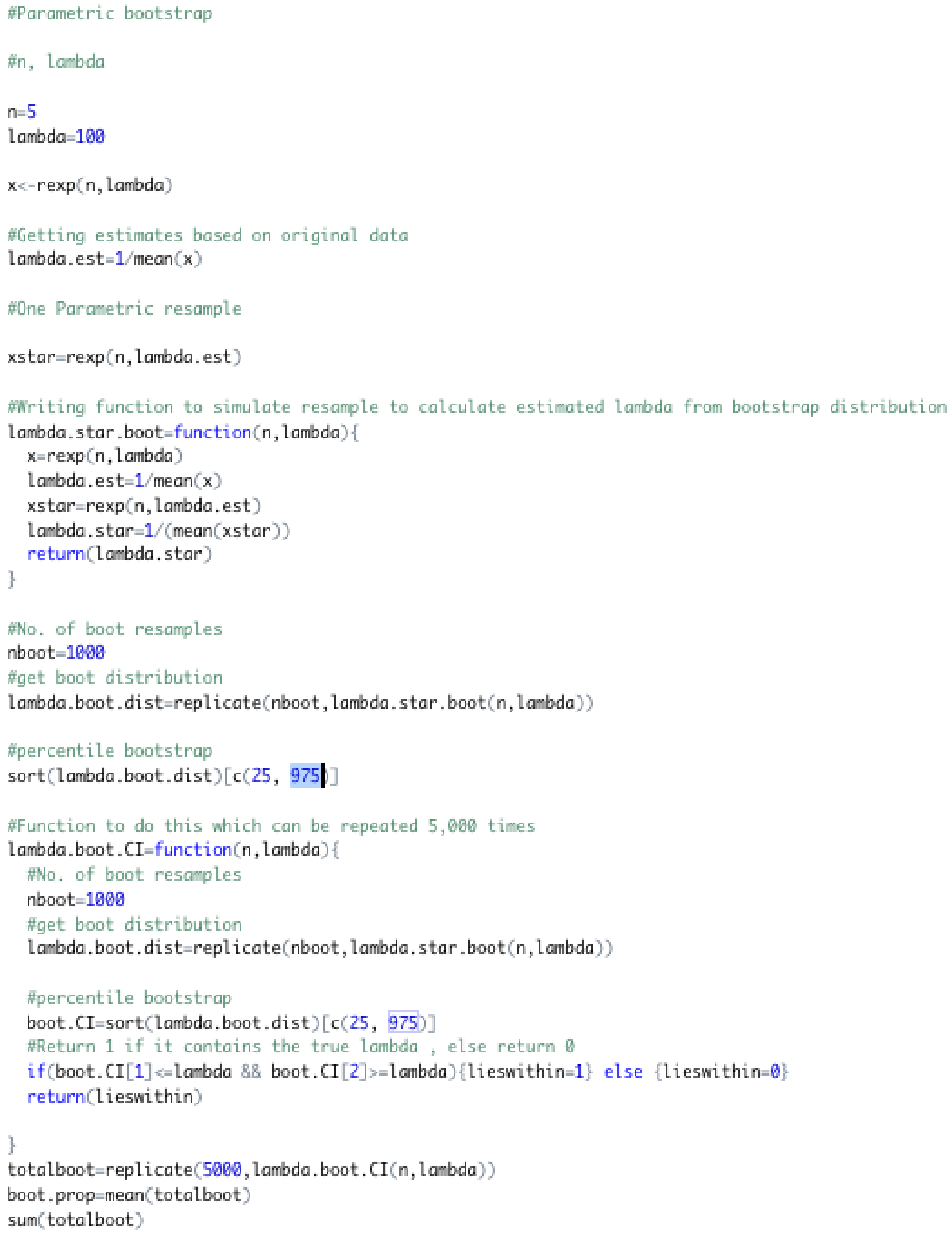


```
# Scatter plots
par(mfrow=c(1,2))

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))
```

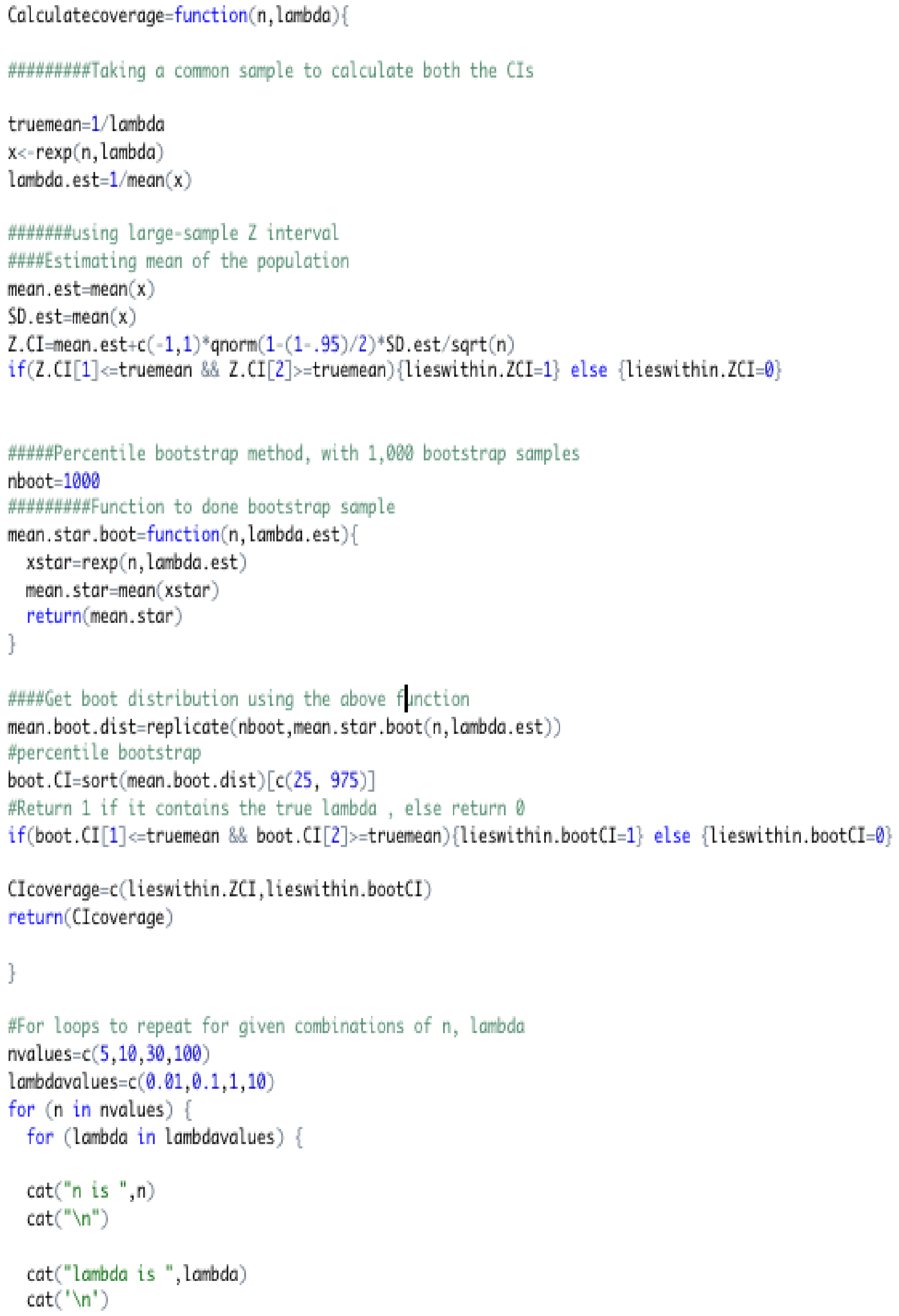
Question 2 Code and Comments



```
#Large sample Z-interval
#Lets assume that the exponontial distribution below is from a large sample z-distribution
n=5
lambda=.1
x<-rexp(n,lambda)
qqnorm(x)
qqline(x)
#Lambda can be estimated by 1/mean(x)
#expected value of 1/mean is lamba
#Expected value of 1/mean is also SD in exponential distribution
lambda.est=1/mean(x)
SD.est=1/mean(x)

CI=lambda.est+c(-1,1)*qnorm(1-(1-.95)/2)*SD.est/sqrt(n)
if(CI[1]<=lambda && CI[2]>=lambda){print(1)} else {print(0)}

#Now writing a function to do this
Z.CI<-function(n, lambda){
  x<-rexp(n,lambda)
  lambda.est=1/mean(x)
  SD.est=1/mean(x)
  CI=lambda.est+c(-1,1)*qnorm(1-(1-.95)/2)*SD.est/sqrt(n)
  #lieswithin is a variable storing the value 1 if CI contains lambda, or 0 if not
  if(CI[1]<=lambda && CI[2]>=lambda){lieswithin=1} else {lieswithin=0}
  return(lieswithin)
}

Z.CI(n,lambda)
totalexp=replicate(5000,Z.CI(n,lambda))
z.total=sum(totalexp)
z.prop=mean(totalexp)
z.prop
```



```
#Parametric bootstrap

#n, lambda

n=5
lambda=100

x<-rexp(n,lambda)

#Getting estimates based on original data
lambda.est=1/mean(x)

#One Parametric resample

xstar=rexp(n,lambda.est)

#Writing function to simulate resample to calculate estimated lambda from bootstrap distribution
lambda.star.boot=function(n,lambda){
  x=rexp(n,lambda)
  lambda.est=1/mean(x)
  xstar=rexp(n,lambda.est)
  lambda.star=1/(mean(xstar))
  return(lambda.star)
}

#No. of boot resamples
nboot=1000
#get boot distribution
lambda.boot.dist=replicate(nboot,lambda.star.boot(n,lambda))

#percentile bootstrap
sort(lambda.boot.dist)[c(25, 975)]

#Function to do this which can be repeated 5,000 times
lambda.boot.CI=function(n,lambda){
  #No. of boot resamples
  nboot=1000
  #get boot distribution
  lambda.boot.dist=replicate(nboot,lambda.star.boot(n,lambda))

  #percentile bootstrap
  boot.CI=sort(lambda.boot.dist)[c(25, 975)]
  #Return 1 if it contains the true lambda , else return 0
  if(boot.CI[1]<=lambda && boot.CI[2]>=lambda){lieswithin=1} else {lieswithin=0}
  return(lieswithin)

}
totalboot=replicate(5000,lambda.boot.CI(n,lambda))
boot.prop=mean(totalboot)
sum(totalboot)
```



```
Calculatecoverage=function(n,lambda){

#########Taking a common sample to calculate both the CIs

truemean=1/lambda
x<-rexp(n,lambda)
lambda.est=1/mean(x)

#######using large-sample Z interval
####Estimating mean of the population
mean.est=mean(x)
SD.est=mean(x)
Z.CI=mean.est+c(-1,1)*qnorm(1-(1-.95)/2)*SD.est/sqrt(n)
if(Z.CI[1]<=truemean && Z.CI[2]>=truemean){lieswithin.ZCI=1} else {lieswithin.ZCI=0}


#####Percentile bootstrap method, with 1,000 bootstrap samples
nboot=1000
#########Function to done bootstrap sample
mean.star.boot=function(n,lambda.est){
  xstar=rexp(n,lambda.est)
  mean.star=mean(xstar)
  return(mean.star)
}

####Get boot distribution using the above function
mean.boot.dist=replicate(nboot,mean.star.boot(n,lambda.est))
#percentile bootstrap
boot.CI=sort(mean.boot.dist)[c(25, 975)]
#Return 1 if it contains the true lambda , else return 0
if(boot.CI[1]<=truemean && boot.CI[2]>=truemean){lieswithin.bootCI=1} else {lieswithin.bootCI=0}

CIcoverage=c(lieswithin.ZCI,lieswithin.bootCI)
return(CIcoverage)

}

#For loops to repeat for given combinations of n, lambda
nvalues=c(5,10,30,100)
lambdavalues=c(0.01,0.1,1,10)
for (n in nvalues) {
  for (lambda in lambdavalues) {

  cat("n is ",n)
  cat("\n")

  cat("lambda is ",lambda)
  cat('\n')
#replicate 5,000 times to calculate the coverage probability according to LLN
coverage.proportion=replicate(5000, Calculatecoverage(n,lambda))

#Each row-one forlarge-sample Z CI coverage and one for parametric bootstrap CI coverage probabilities.
cat("The coverage probabilities for the large-sample Z CI and parametric bootstrap CI respectively are:")
#Means of each row (first for coverage prob. of large-sample Z CI
#and second for the percentile bootstrap CI coverage probability
meancoverage=rowMeans(coverage.proportion)
cat("\n",meancoverage)
cat("\n")
cat("\n")

zprop=coverage.proportion[c(TRUE,FALSE)]
zprop
bootprop=coverage.proportion[c(FALSE,TRUE)]
mean(zprop)
mean(bootprop)
}
}

plotthecoverage<-function(n,height,ylimit){

  collectiveresultstrial1<-c()
  for(lambda in c(0.01,0.1,1,10)){
    coverage.proportion=replicate(5000, Calculatecoverage(n,lambda))
    meancoverage=rowMeans(coverage.proportion)
    collectiveresultstrial1<-c(collectiveresultstrial1,meancoverage)
  }
  #plots for both meanerrortheta1 and meanerrortheta2
  lambda<-c(0.01,0.1,1,10)
  # coverageofz stores the values of coverage probabilities for large-sample Z CI,  #values are separated from all results of estimators using the line below.
  #Alternate values of the results are collected from vectors starting from #element 1
  coverageofz<-collectiveresultstrial1[c(TRUE,FALSE)]

  # coverageofboot stores the values of coverage probabilities for the percentile bootstrap CI#are separated from all results of estimators using the line below.
  #Alternate values of the results are collected from vector starting from #element 2
  coverageofboot<-collectiveresultstrial1[c(FALSE,TRUE)]

  #Plots
  #Coverage of large-sample Z CI(Blue line)
  plot(x=lambda,y=coverageofz,type="b",pch=18, col="blue",lty=2,ylab="Coverage probability",xlab="Lambda",ylim = ylimit)

  #Coverage of percentile boot CI
  lines(x=lambda,y=coverageofboot,type="b",pch=19,col="red")


  #height is just a variable to position the legend
  legend(1, height, legend=c( "Percentile bootstrap CI","Large-sample Z CI"),
         col=c("red", "blue"), lty=1:2, cex=0.8)
}
```