Mini Project #5

Tanushri Singh, Nikhil Pareek

Contribution of each group member:

Nikhil worked on problem number 1 and Tanu worked on questions 2 then went onto Document questions and report all the findings. Both partners worked efficiently to complete the project requirements!

NOTE: ALL CODES ARE ATTACHED IN SECTION 2!

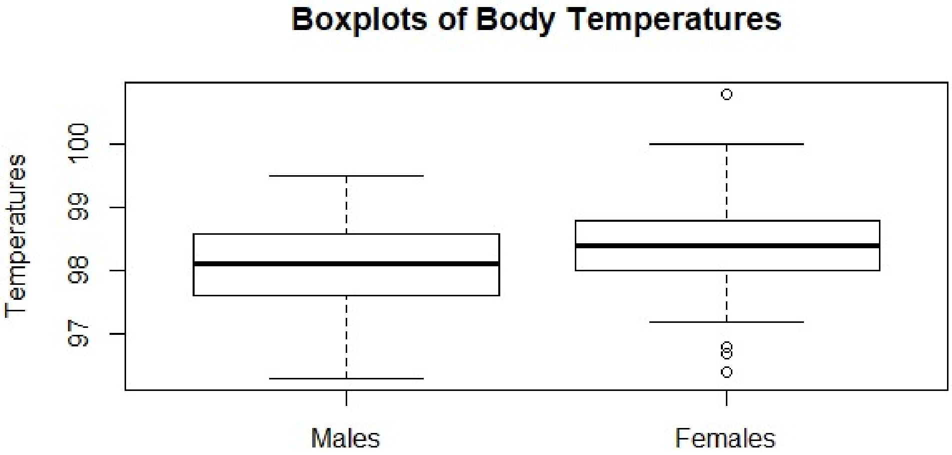
Section 1

1. **Consider the data stored in bodytemp-heartrate.csv on eLearning, containing measurements of body temperature and heart rate for 65 male (gender = 1) and 65 female (gender = 2) subjects.**
   (Refer Section 2 for R code)

   First we use the read.csv function to read the csv data into the variable.
   Syntax: read.csv(file, separator, header…)
   Here: file – file location
   Separator – separators like '.', '.' Etc
   Header – takes a Boolean value (yes, no). Specifies whether the file has header

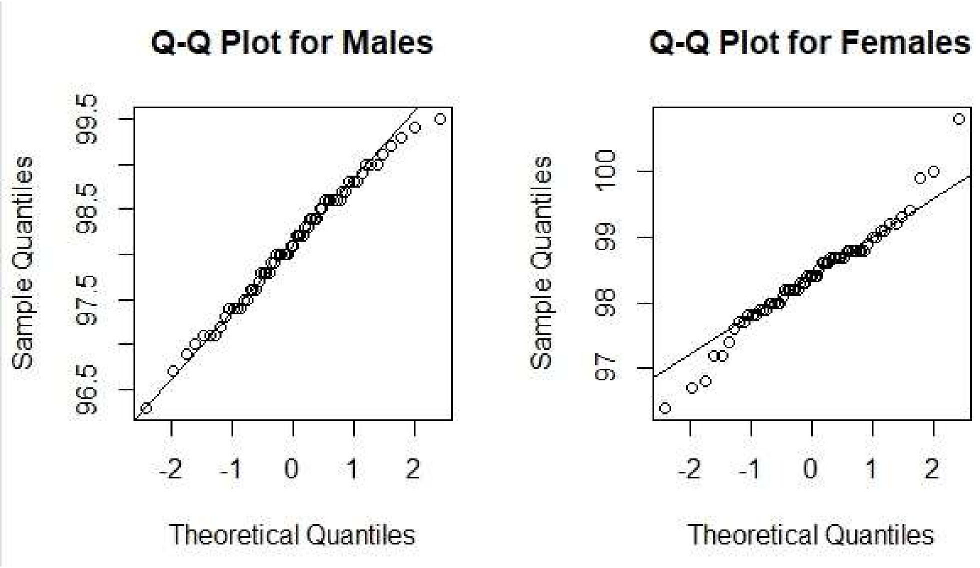
   Now separate the two data sets by using the subset function which returns subsets of vectors and matrices.
   Syntax: subset(x, …)
   Here: x – data set to be subsetted
   We can follow this data set with any relational condition we need to separate the data set on

   a. **Do males and females differ in mean body temperature? Answer this question by performing an appropriate analysis of the data, including an exploratory analysis.**
   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 $\bar{m}$ estimates the population mean μ_m and the sample mean $\bar{f}$ estimates the population mean μ_f

We take the null hypothesis H_0 : Difference between means = 0 => $\bar{m} - \bar{f} = 0$
And Alternate Hypothesis H_1 : Difference between means ≠0 => $\bar{m} - \bar{f} \neq 0$

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

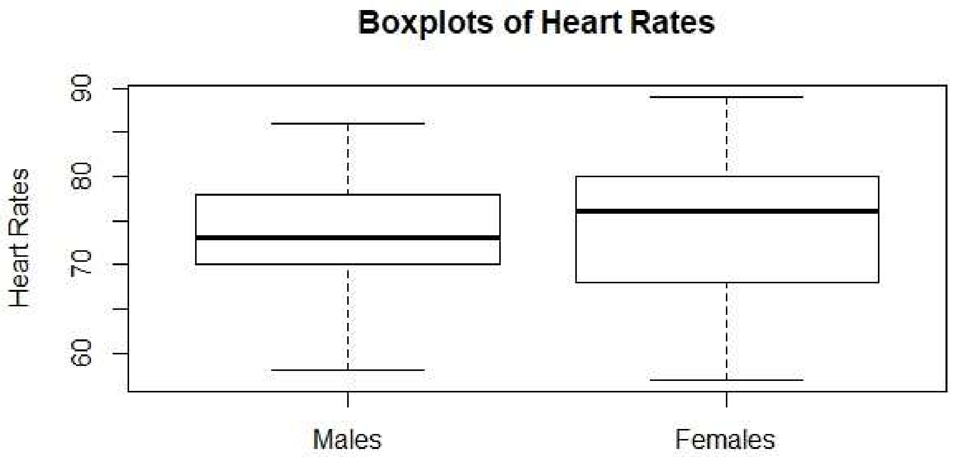
We construct the confidence interval using t.test function in R
Syntax: t.test(x, alternative, paired, conf.level, var.equal…)
Here: x – data set
alternative – specify whether the alternate hypothesis is two-sided or not
paired – Boolean value to specify whether the two samples are paired or independent
conf.level – confidence level of the interval required
var.equal – specify whether the population variances are same or not

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.

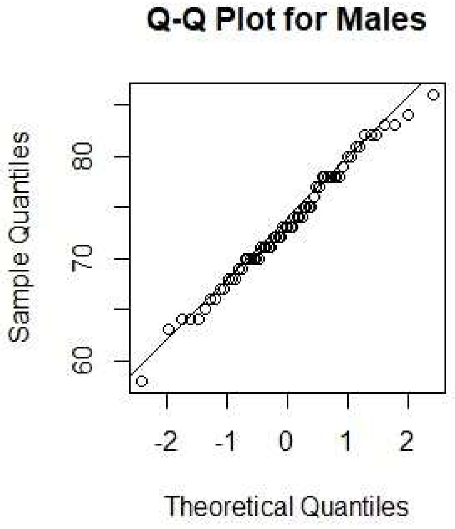
b. Do males and females differ in mean heart rate? Answer this question by performing an appropriate analysis of the data, including an exploratory analysis.

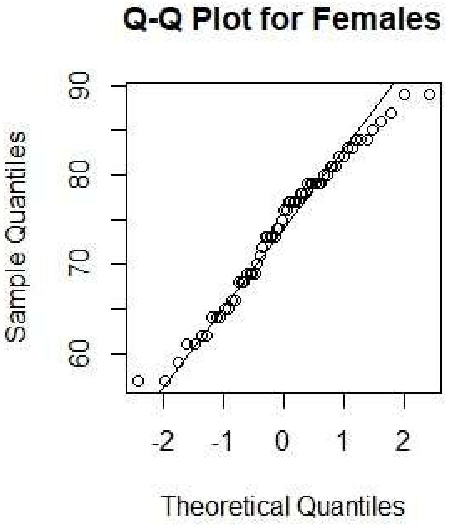
Let’s draw the boxplots for the heart rate values for both females and males



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.

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 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 $\overline{m}$ estimates the population mean μ_m and the sample mean $\bar{f}$ estimates the population mean μ_f

We take the null hypothesis H_0 : Difference between means = 0 => $\bar{m} - \bar{f} = 0$
And Alternate Hypothesis H_1 : Difference between means ≠0 => $\bar{m} - \bar{f} \neq 0$

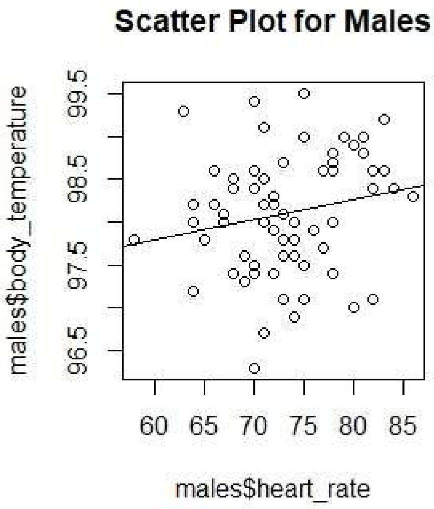
The samples here are to be treated as independent samples, with unequal variances coming from an approximately normal distribution, hence we can use t-distribution with Satterthwaite's approximation to get the confidence interval

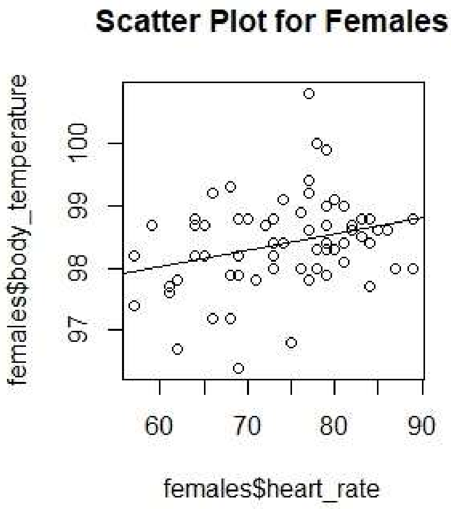
We construct the confidence interval using t.test function in R
Syntax: t.test(x, alternative, paired, conf.level, var.equal…)
Here: x – data set
alternative – specify whether the alternate hypothesis is two-sided or not
paired – Boolean value to specify whether the two samples are paired or independent
conf.level – confidence level of the interval required
var.equal – specify whether the population variances are same or not

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. **Is there a linear relationship between body temperature and heart rate? Does this relationship depend on gender? Answer these questions by performing an appropriate analysis of the data, including an exploratory analysis.**
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.

Syntax: cor(x, y…)
Here: x, y are datasets or variables.

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

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

2. **The goal of this exercise to see how large n should be for the large-sample and the (parametric) bootstrap percentile method confidence intervals for the mean of an exponential population to be accurate. To be specific, let X_1,...,X_n represent a random sample from an exponential (λ) distribution. Note that this distribution is skewed and its mean is μ = 1/λ. We can construct two confidence intervals for μ — one the large-sample z-interval (interval 1) and the other a (parametric) bootstrap percentile method interval (interval 2). We would like to investigate their accuracy, i.e., how close their estimated coverage probabilities are to the assumed nominal level of confidence, for various combinations of (n,λ). This investigation will focus on 1−α = 0.95, λ = 0.01, 0.1, 1, 10 and n = 5, 10, 30, 100. Thus, we have a total of 4∗4 = 16 combinations of (n,λ) to investigate.**
   (Refer Section 2 for R Code)

   a. **For a given setting, compute Monte Carlo estimates of coverage probabilities of the two intervals by simulating appropriate data, using them to construct the two confidence intervals, and repeating the process 5000 times.**
   To simulate Monte Carlo estimates of coverage probabilities and to construct the confidence intervals, we have created the following functions:
   checkzci – takes n and λ 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 λ 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 λ 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 λ as input parameters, constructs a parametric initial bootstrap sample and calls checkbci 5000 times and calculates the coverage probabilities

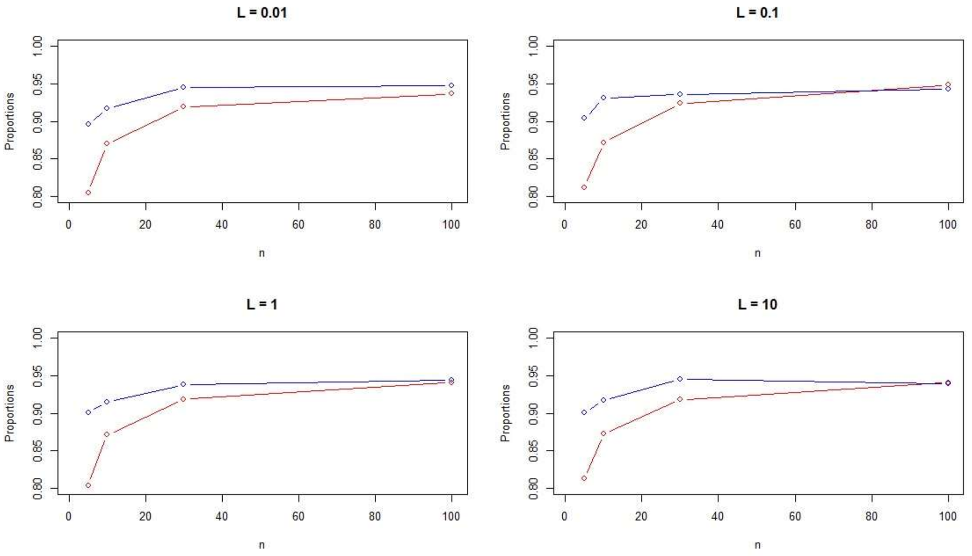
   Using these functions, for the (n,λ) combination as (5, 0.01) we get the coverage probabilities as:
   Z-interval: 0.8056
   Bootstrap interval: 0.8960

   b. **Repeat (a) for the remaining combinations of (n, λ). Present an appropriate summary of the results.**
   Repeating the above process for the remaining combinations we get:

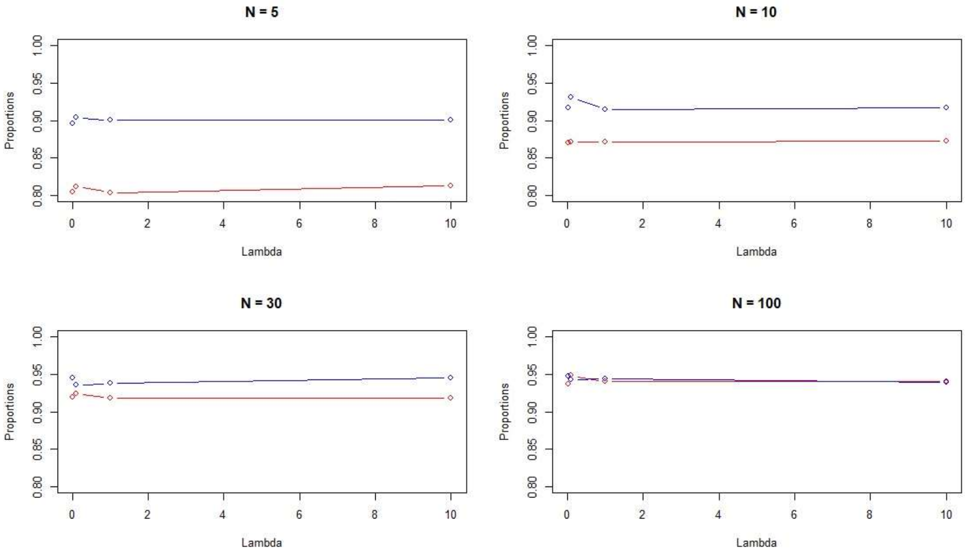
Z-proportions	L = 0.01	L = 0.1	L = 1	L = 10
N = 5	0.8056	0.8124	0.8042	0.8132
N = 10	0.8702	0.8716	0.8716	0.8728
N = 30	0.9192	0.9236	0.9184	0.9178
N = 100	0.9366	0.9482	0.9404	0.9408

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 λ fixed



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

c. Interpret all the results. Be sure to answer the following questions: In case of the large-sample interval, how large n is needed for the interval to be accurate? Likewise, in case of the bootstrap interval, how large n is needed for the interval to be accurate? Do these answers depend on λ? Can we say that one method is more accurate than the other? Which interval would you recommend? Provide justification for all your conclusions.

From Graph 1, we see that the graphs don't change drastically when λ is changed, so we can say that the coverage probabilities don't depend on λ. And we also see that the coverage probabilities we get via bootstrap are higher than those of zinterval method. From Graph 2, 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) For the bootstrap method coverage probabilities, they are on the higher side (approximately) from n=30 onwards Taking into account 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, hence bootstrap method is more accurate even for the low values of n. Hence bootstrap ,method is recommended

(d) Do your conclusions in (c) depend on the specific values of λ that were fixed in advance? Explain.

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 0.61
The coverage probability for large sample z for n 5 lambda 0.1 is 0.8058
```

And,

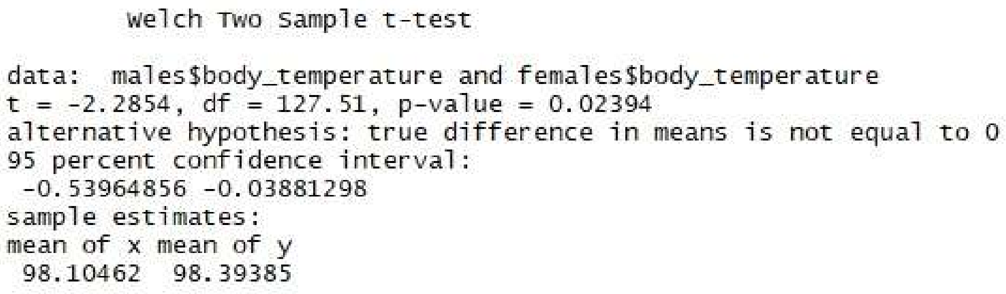
```
The coverage probability for bootstrap is for n 10 lambda 0.1 is 0.695
The coverage probability for large sample z for n 10 lambda 0.1 is 0.8758
The coverage probability for bootstrap is for n 30 lambda 0.1 is 0.7134
The coverage probability for large sample z for n 30 lambda 0.1 is 0.9114
The coverage 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

Section 2

R code for Question 1

```
#reading data using read.csv function
	> bodytemp_heartrate = read.csv(bodytemp-heartrate.csv, header = T )
#separating the two datasets
	> males = subset(bodytemp_heartrate, bodytemp_heartrate$gender == 1)
	> females = subset(bodytemp_heartrate, bodytemp_heartrate$gender == 2)
#drawing boxplots for body temperature values
	> boxplot(males$body_temperature, females$body_temperature, main = "Boxplots of Body Temperatures", names = c('Males', 'Females'), ylab = "Temperatures")
#drawing Q-Q plots for the body temperature values
	> 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)
#confidence interval using t,test function for the body temperature values
	> t.test(males$body_temperature, females$body_temperature, alternative = 'two.sided', var.equal = F)
```

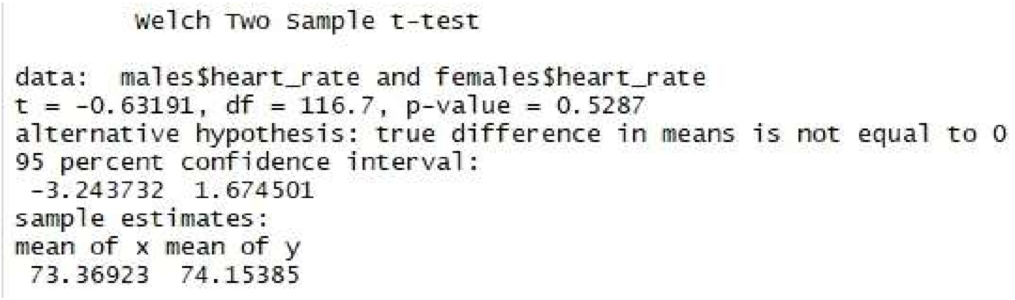


```
        Welch Two Sample t-test

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 
```

```
#drawing boxplot for the heart rate values
	> boxplot(males$heart_rate, females$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(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)
```

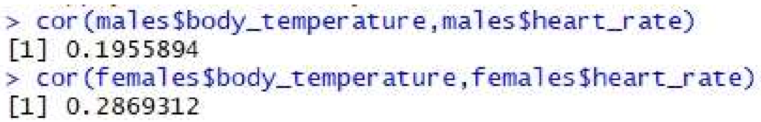
#getting the confidence interval using the t.test function

```
> 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
```

#finding the correlation values between body temperatures and heart rates



```
> cor(males$body_temperature,males$heart_rate)
[1] 0.1955894
> cor(females$body_temperature,females$heart_rate)
[1] 0.2869312
```

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

```
> 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))
```

R code for Question 2

```
#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] 0.8056
#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] 0.8960
#generating the proportion values for bootstrap and z-interval for all the combinations of n
and λ
        > 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')
```