Mini project #4

Group Member: Chaoran Li, Wenting Wang

Contribution of each member:

Firstly, we discussed the mathematical models and code details together. Then, we divided the project into two part and finished our respective work. Wenting Wang mainly worked on Q1 while Chaoran Li worked on Q2. Then, we merged our code and solution into one report. Each member makes contribution to each sub task of this project and combines all to finish this project, as the details shown in table 1.

	Question1	Question2
Chaoran li	20%	20%
Wenting wang	80%	80%

Table 1: Member contribution table

Question 1:

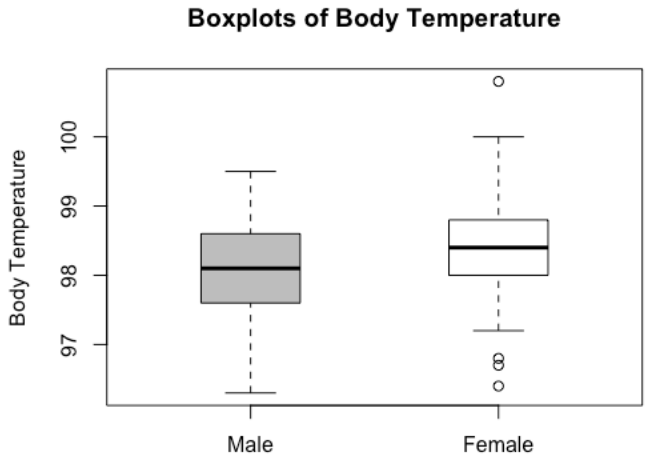
(a) First, we set up the Null Hypothesis and Alternate Hypothesis:

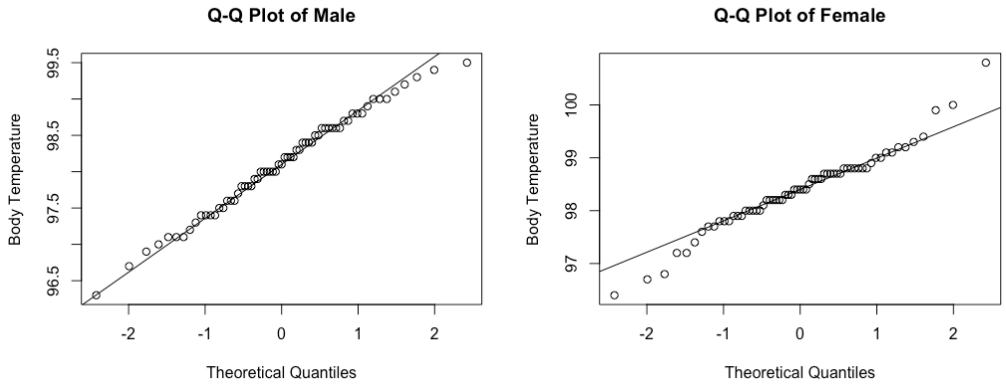
H_0: No difference of mean temperature between male and female: $Tu_M - Tu_F = 0$

H_1: There' s difference of mean temperature between male and female: $Tu_M - Tu_F \neq 0$

Next, we analyze the data:

The body temperatures of male and female are independent sample data, and the population standard deviation unknown. Based on the boxplots and QQ-plots below we can see the variances are different and the datasets are approximately normal.





Thus, we choose to use t-test and get the results below:

```
        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
```

Then we can conclude that because the 95% confidence interval of difference between male and female temperature is [-0.53964856 -0.03881298], which does not include 0. The P-value is 0.02394 which is less than 0.05. Thus, we reject the null hypothesis, which means there exists difference of mean temperature between male and female. And from the sample means, we can notice that the mean temperature of female is a little higher than male.

Code:

```r
# (a) Body temperature
bodytemp.heartrate <- read.csv(
  file=file.path("./Mini Project 5/bodytemp-heartrate.csv"))
male <-bodytemp.heartrate[which(bodytemp.heartrate$gender==1),]
female <- bodytemp.heartrate[which(bodytemp.heartrate$gender==2),]
#plots body temperature
boxplot(male$body_temperature, female$body_temperature, boxwex=.4,
        main = "Boxplots of Body Temperature", names = c('Male', 'Female'),
        col=c("grey", "white"), ylab="Body Temperature")
qqnorm(male$body_temperature, main='Q-Q Plot of Male', ylab="Body Temperature")
qqline(male$body_temperature)
qqnorm(female$body_temperature, main='Q-Q Plot of Female',
                                                    ylab="Body Temperature")
qqline(female$body_temperature)
#t,test function for the body temperature difference
t.test(male$body_temperature, female$body_temperature,
       alternative ='two.sided', var.equal = F)
```

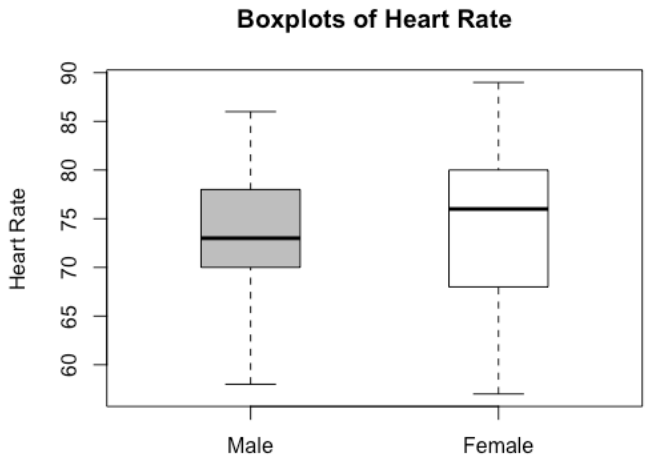
(b) First, we set up the Null Hypothesis and Alternate Hypothesis:

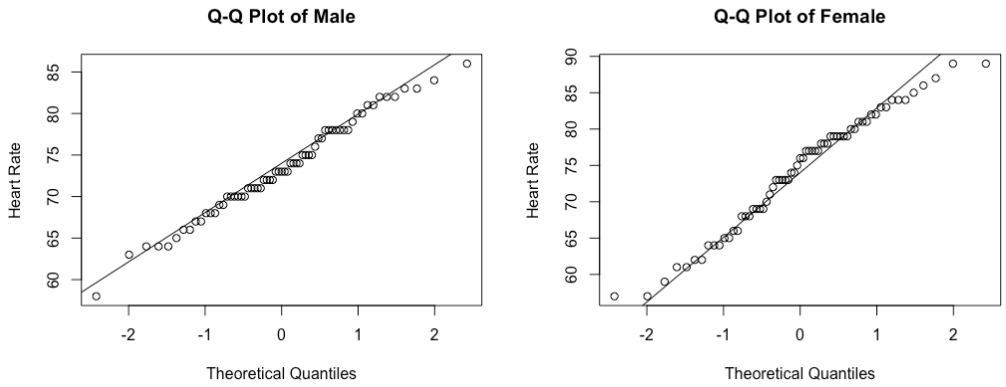
H_0: No difference of mean heart rate between male and female: $Hu_M - Hu_F = 0$

H_1: There' s difference of mean heart rate between male and female: $Hu_M - Hu_F \neq 0$

Next, we analyze the data:

The heart rates of male and female are independent sample data, and the population standard deviation unknown. Based on the boxplots and QQ-plots below we can see the variances are different and the datasets are approximately normal.





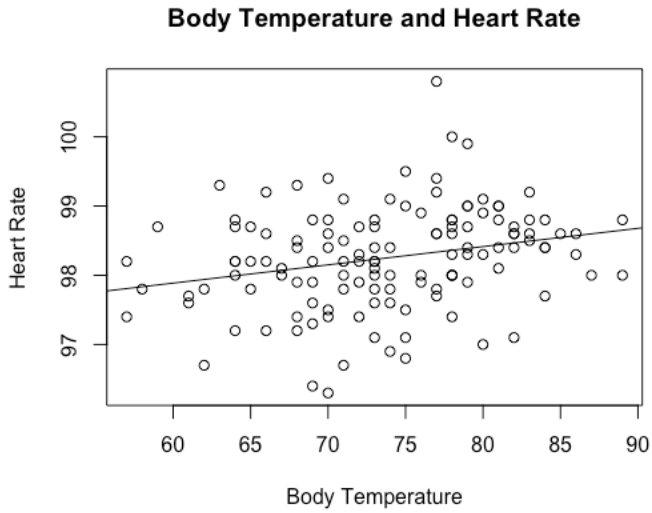

```
        Welch Two Sample t-test

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
```

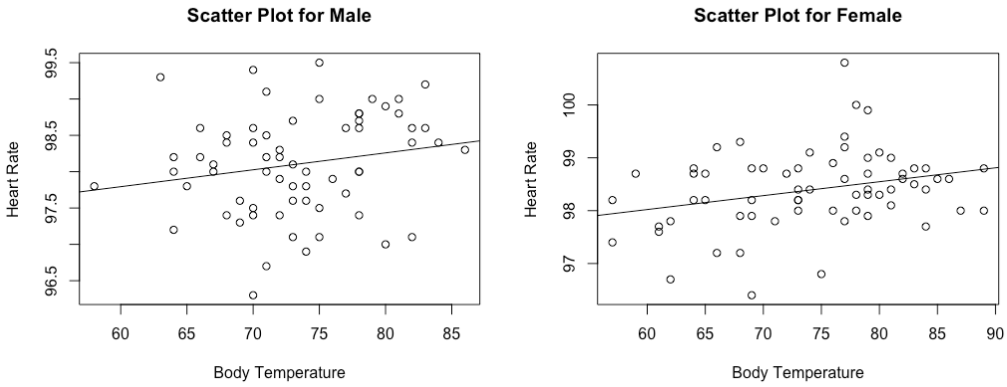
Then we can conclude that because the 95% confidence interval of difference between male and female heart rate is [-3.243732 1.674501], which does include 0. The P-value is 0.5287 which is greater than 0.05. Thus, we accept the null hypothesis, which means there is no difference of mean heart rates between male and female.

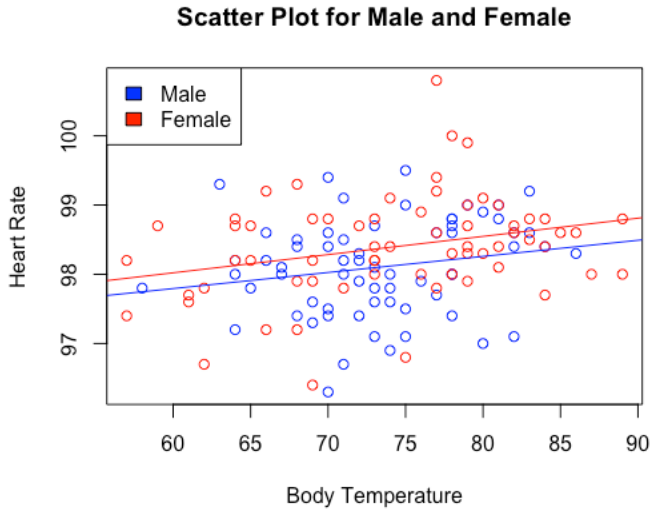
Code:

```
# (b) Heart rate
boxplot(male$heart_rate, female$heart_rate, boxwex=.4,
        main = "Boxplots of Heart Rate", names = c('Male', 'Female'),
        col=c("grey", "white"), ylab="Heart Rate")
qqnorm(male$heart_rate, main="Q-Q Plot of Male", ylab="Heart Rate")
qqline(male$heart_rate)
qqnorm(female$heart_rate, main="Q-Q Plot of Female", ylab="Heart Rate")
qqline(female$heart_rate)
#t,test function for the body temperature difference
t.test(male$heart_rate, female$heart_rate,
       alternative='two.sided', var.equal=F)
```

(c) First, we draw the scatter plot and regression line below







We can see that in both the plot of all people and the plots of male and female, there exists a weak positive trend between body temperature and heart rate. We can also see some outliers in the scatter plots. Then we can calculate the correlation between body temperature and heart rate in the 3 plots.

Cor(all) = 0.2536564, Cor(male) = 0.1955894, Cor(female) = 0.2869312

We can also get the simple linear regression as below:

```
> #get the fitted regression line
> lm(bodytemp.heartrate$body_temperature ~ bodytemp.heartrate$heart_rate)

Call:
lm(formula = bodytemp.heartrate$body_temperature ~ bodytemp.heartrate$heart_rate)

Coefficients:
                  (Intercept)  bodytemp.heartrate$heart_rate
                     96.30675                        0.02633

> lm(male$body_temperature ~ male$heart_rate)

Call:
lm(formula = male$body_temperature ~ male$heart_rate)

Coefficients:
    (Intercept)  male$heart_rate
       96.39789          0.02326

> lm(female$body_temperature ~ female$heart_rate)

Call:
lm(formula = female$body_temperature ~ female$heart_rate)

Coefficients:
      (Intercept)  female$heart_rate
         96.44211            0.02632
```

Above all, we can conclude that no matter male or female there is weak positive relationship between body temperature and heart rate. But the correlation of female is a little stronger than male.

Code:

```r
# (c) Compare male and female
plot(x=bodytemp.heartrate$heart_rate, y=bodytemp.heartrate$body_temperature,
     type="p", xlab="Body Temperature", ylab="Heart Rate",
     main="Body Temperature and Heart Rate")
abline(lm(bodytemp.heartrate$body_temperature ~ bodytemp.heartrate$heart_rate))
cor(bodytemp.heartrate$body_temperature, bodytemp.heartrate$heart_rate)
#get the fitted regression line
lm(bodytemp.heartrate$body_temperature ~ bodytemp.heartrate$heart_rate)
#abline(temp)

#Scatter plots for body temperature and heart rate for males
plot(male$heart_rate, male$body_temperature,
     type="p", xlab="Body Temperature", ylab="Heart Rate",
     main="Scatter Plot for Male")
abline(lm(male$body_temperature ~ male$heart_rate))
cor(male$body_temperature, male$heart_rate)
lm(male$body_temperature ~ male$heart_rate)
```

```r
# for females
plot(female$heart_rate, female$body_temperature, pch=1,
     type="p", xlab="Body Temperature", ylab="Heart Rate",
     main="Scatter Plot for Female")
abline(lm(female$body_temperature ~ female$heart_rate))
cor(female$body_temperature, female$heart_rate)
lm(female$body_temperature ~ female$heart_rate)
# mixed scatter plots
plot(bodytemp.heartrate$heart_rate, bodytemp.heartrate$body_temperature,
     col=c('blue', 'red')[unclass(bodytemp.heartrate$gender)],
     type="p", xlab="Body Temperature", ylab="Heart Rate",
     main="Scatter Plot for Male and Female")
legend("bottomright", c("Male", "Female"), fill=c('blue', 'red'))
abline(lm(male$body_temperature ~ male$heart_rate), col='blue')
abline(lm(female$body_temperature ~ female$heart_rate), col='red')
```

Question 2:

Compare large-sample z-interval (interval 1) and bootstrap percentile method interval (interval 2) from an exponential distribution.

(a) Given $\boldsymbol{\lambda}$ = 0.1 and n = 30, compute two accuracies of two intervals which confidence interval include the mean.

Firstly, we design two functions to return the result whether the confidence interval include the mean in one investigation.

For interval 1, even though we could know the distribution belongs to exponential distribution, the standard deviation or variance are unknown. Hence, we have to use the sample' s standard deviation or variance. Besides, even though n = 5 or 10 is not enough for a large-sample assumption, we will still use qnorm() for getting large-sample z-interval.

```r
# Question 2
# Large-sample z-interval and bootstrap percentile method interval.

# (a) Given lambda = 0.1 and n = 30, compare two intervals' accuracy.
# 1 - a = 0.95 => [0.025, 0.975].

# Large-sample z-interval
getInterval1 <- function(n, lambda) {
  population.mean = 1/lambda
  sample = rexp(n, rate=lambda)
  center = mean(sample)
  margin = qnorm(0.975) * sd(sample) / sqrt(n)
  if (center - margin > population.mean) {
    return (0)
  } else if (center + margin < population.mean) {
    return (0)
  } else {
    return (1)
  }
}
```

For interval 2, we know the distribution is exponential distribution. Hence, we would use parametric bootstrap here.

```r
# Bootstrap percentile method interval
getInterval2 <- function(n, lambda) {  #
  population.mean = 1/lambda
  sample = rexp(n, rate=lambda)
  lambda.est = 1 / mean(sample)
  sample.boot = c(mean(sample),
          replicate(999, expr=mean(rexp(n, rate=lambda.est)))) # 1000 times
  sample.boot = sort(sample.boot)
  if (sample.boot[25] > population.mean) {
    return (0)
  } else if (sample.boot[975] < population.mean) {
    return (0)
  } else {
    return (1)
  }
}
```

Secondly, we use the given parameters to calculate the two accuracy of two intervals during repeating 5000 times.

```r
q2.round = 5000
q2.a.lambda = 0.1
q2.a.n = 30
q2.a.interval1s = replicate(q2.round, expr=getInterval1(q2.a.n, q2.a.lambda))
q2.a.accuracy1 = sum(q2.a.interval1s) / q2.round
q2.a.interval2s = replicate(q2.round, expr=getInterval2(q2.a.n, q2.a.lambda))
q2.a.accuracy2 = sum(q2.a.interval2s) / q2.round
sprintf("Lambda = %f, n = %d. Accuracy:", q2.a.lambda, q2.a.n)
sprintf("interval 1: %f; interval 2 %f.", q2.a.accuracy1, q2.a.accuracy2)
```

Then, we get the results below:

```
> q2.round = 5000
> q2.a.lambda = 0.1
> q2.a.n = 30
> q2.a.interval1s = replicate(q2.round, expr=getInterval1(q2.a.n, q2.a.lambda))
> q2.a.accuracy1 = sum(q2.a.interval1s) / q2.round
> q2.a.interval2s = replicate(q2.round, expr=getInterval2(q2.a.n, q2.a.lambda))
> q2.a.accuracy2 = sum(q2.a.interval2s) / q2.round
> sprintf("Lambda = %f, n = %d. Accuracy:", q2.a.lambda, q2.a.n)
[1] "Lambda = 0.100000, n = 30. Accuracy:"
> sprintf("interval 1: %f; interval 2 %f.", q2.a.accuracy1, q2.a.accuracy2)
[1] "interval 1: 0.915200; interval 2 0.935600."
```

The accuracy of interval 2 which is 0.9356 is larger than that of interval 1 which is 0.9152.

(b) Repeat (a) for the remaining combinations of n∈ {5, 10, 30, 100} and $\boldsymbol{\lambda} \in$ {0.01, 0.1, 1. 10}.

```
# (b) Traverse all lambdas and ns.
q2.b.lambdas = c(0.01, 0.1, 1, 10)
q2.b.ns = c(5, 10, 30, 100)
q2.intervals = c("interval 1", "interval 2")
q2.b.accuracies = array(0,
        dim=c(length(q2.b.lambdas), length(q2.b.ns), length(q2.intervals)))
for (i in 1:length(q2.b.lambdas)) {
  lambda = q2.b.lambdas[i]
  for (j in 1:length(q2.b.ns)) {
    n = q2.b.ns[j]
    interval1s = replicate(q2.round, expr=getInterval1(n, lambda))
    accuracy1 = sum(interval1s) / q2.round
    q2.b.accuracies[i, j, 1] = sum(interval1s) / q2.round
    interval2s = replicate(q2.round, expr=getInterval2(n, lambda))
    accuracy2 = sum(interval2s) / q2.round
    q2.b.accuracies[i, j, 2] = sum(interval2s) / q2.round
  }
}
print(q2.b.accuracies)
```

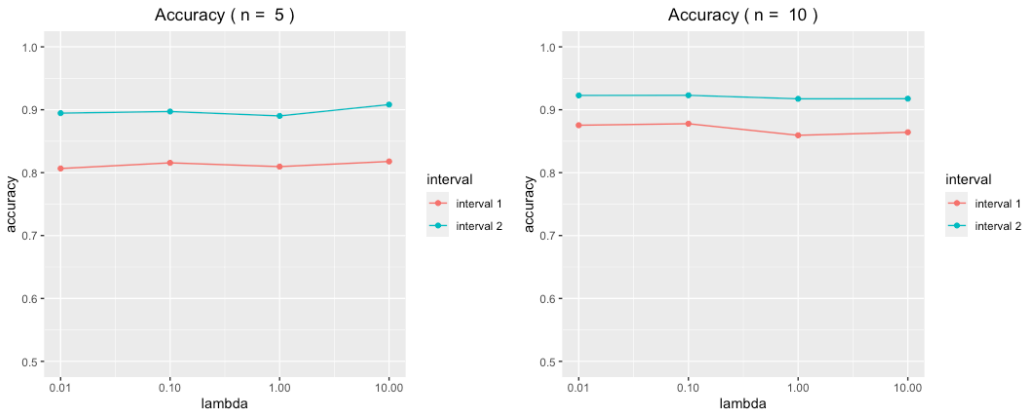
```
> print(q2.b.accuracies)
, , 1

       [,1]   [,2]   [,3]   [,4]
[1,] 0.8066 0.8754 0.9152 0.9424
[2,] 0.8156 0.8778 0.9132 0.9386
[3,] 0.8096 0.8594 0.9182 0.9356
[4,] 0.8178 0.8642 0.9194 0.9414

, , 2

       [,1]   [,2]   [,3]   [,4]
[1,] 0.8946 0.9228 0.9436 0.9514
[2,] 0.8972 0.9230 0.9372 0.9472
[3,] 0.8902 0.9174 0.9394 0.9454
[4,] 0.9082 0.9176 0.9416 0.9456
```

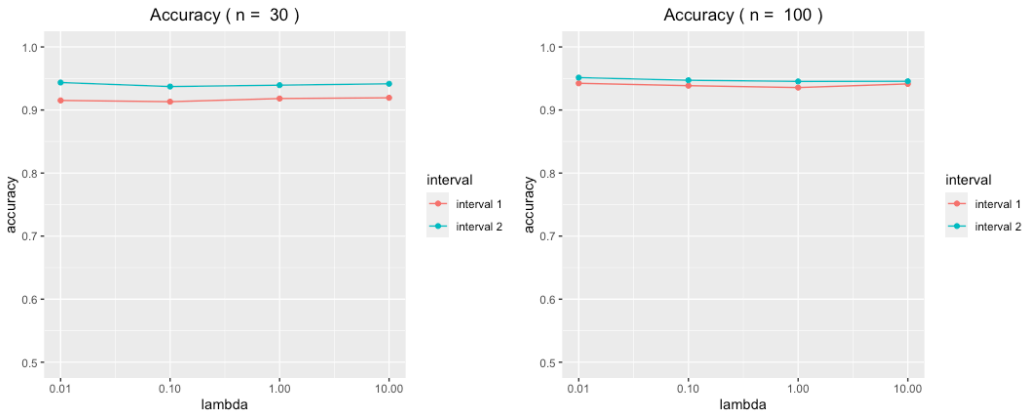
For $\boldsymbol{\lambda} \in$ {0.01, 0.1, 1, 10}, it is unclear to show the result in regular scale. So, I use ggplot2 library to make x coordinate log scale when talking about different $\boldsymbol{\lambda}$.

```r
library(ggplot2)
# Different lambdas and one n.
for (j in 1:length(q2.b.ns)) {
  df = data.frame(
    lambda = rep(q2.b.lambdas, times=length(q2.intervals)),
    interval = rep(q2.intervals, each=length(q2.b.lambdas)),
    accuracy = c(q2.b.accuracies[, j, ]))
  img = ggplot(data=df, mapping=aes(x=lambda, y=accuracy, colour=interval)) +
    geom_line() + geom_point() + # line plus scatter
    scale_x_continuous(trans='log10') + # log scale in x coordinate
    ylim(0.5, 1) + # y range: [0, 1]
    labs(title=paste("Accuracy ( n = ", q2.b.ns[j], ")")) + # add title
    theme(plot.title = element_text(hjust = 0.5)) # center the title
  print(img)
}
# One lambda and different ns.
for (i in 1:length(q2.b.lambdas)) {
  df = data.frame(
    n = rep(q2.b.ns, times=length(q2.intervals)),
    interval = rep(q2.intervals, each=length(q2.b.ns)),
    accuracy = c(q2.b.accuracies[i, , ]))
  img = ggplot(data=df, mapping=aes(x=n, y=accuracy, colour=interval)) +
    geom_line() + geom_point() + # line plus scatter
    ylim(0.5, 1) + # y range: [0, 1]
    labs(title=paste("Accuracy ( lambda = ", q2.b.lambdas[i], ")")) + # add titl
    theme(plot.title = element_text(hjust = 0.5)) # center the title
  print(img)
}
```

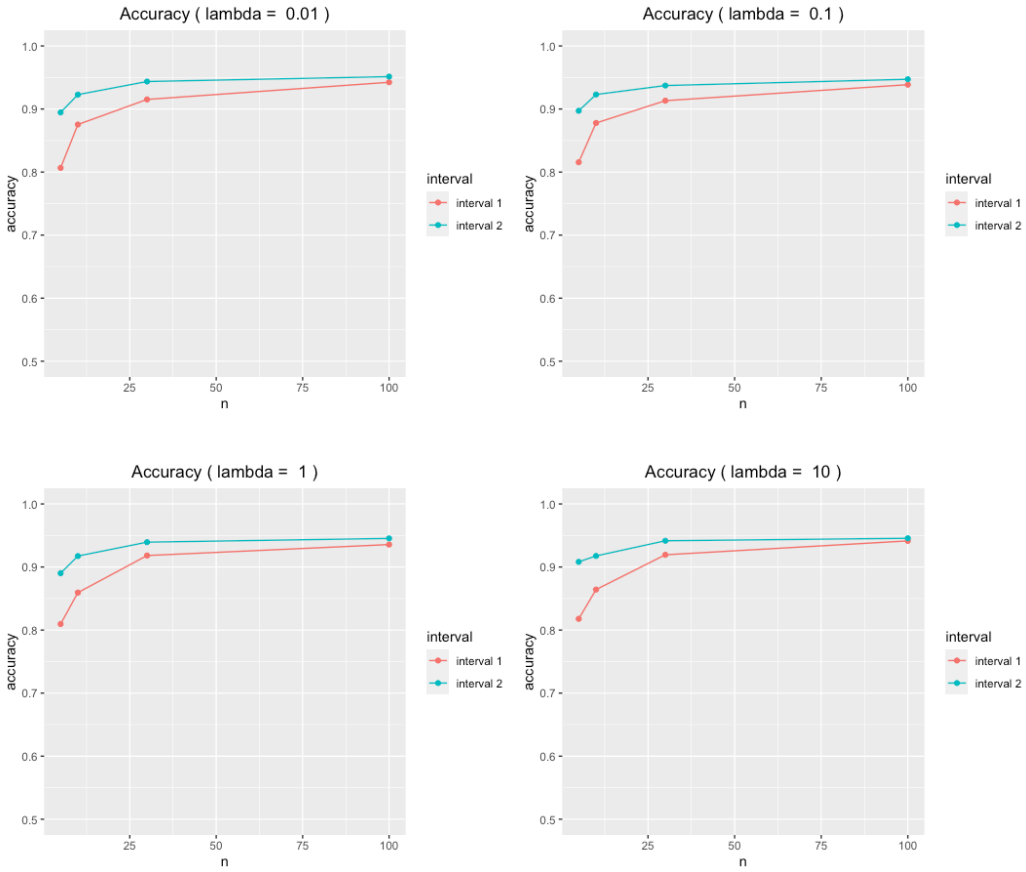
Then, we got the diagrams below:

Discuss the influence of **λ** with each n.





Discuss the influence of n with each **λ**.



From the eight diagrams above, we can make a brief summary now. The accuracy is independent from **λ**. While n increases, the accuracy increases and approaches convergence.

(c) Answer the following questions:

(1) How large n is needed for the large-sample z-interval?

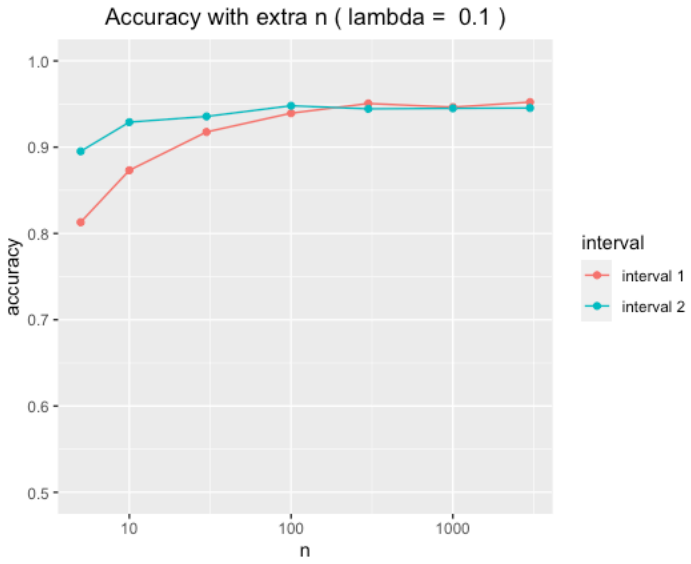
n = 300 is large enough for the large-sample z-interval.

From the first four diagrams in (b), we can see that **λ** has no effect on its accuracy. However,

we cannot make sure whether the function converges at n = 100 in interval 1. I tried some bigger n below:

```
# (c) Analyse the result we got in (b)
q2.c.lambda = 0.1
q2.c.ns = c(5, 10, 30, 100, 300, 1000, 3000)
q2.c.accuracies = array(0,
          dim=c(length(q2.c.ns), length(q2.intervals)))
for (i in 1:length(q2.c.ns)) {
  n = q2.c.ns[i]
  interval1s = replicate(q2.round, expr=getInterval1(n, q2.c.lambda))
  accuracy1 = sum(interval1s) / q2.round
  q2.c.accuracies[i, 1] = sum(interval1s) / q2.round
  interval2s = replicate(q2.round, expr=getInterval2(n, q2.c.lambda))
  accuracy2 = sum(interval2s) / q2.round
  q2.c.accuracies[i, 2] = sum(interval2s) / q2.round
}
print(q2.c.accuracies)

q2.c.df = data.frame(
  n = rep(q2.c.ns, times=length(q2.intervals)),
  interval = rep(q2.intervals, each=length(q2.c.ns)),
  accuracy = c(q2.c.accuracies[, ]))
ggplot(data=q2.c.df, mapping=aes(x=n, y=accuracy, colour=interval)) +
  geom_line() + geom_point() + # line plus scatter
  scale_x_continuous(trans='log10') + # log scale in x coordinate
  ylim(0.5, 1) + # y range: [0, 1]
  labs(title=paste(
    "Accuracy with extra n ( lambda = ", q2.c.lambda, ")")) + # add title
  theme(plot.title = element_text(hjust = 0.5)) # center the title
```

```
> print(q2.c.accuracies)
       [,1]   [,2]
[1,] 0.8130 0.8952
[2,] 0.8732 0.9290
[3,] 0.9176 0.9356
[4,] 0.9394 0.9480
[5,] 0.9506 0.9444
[6,] 0.9464 0.9450
[7,] 0.9522 0.9454
```



From the diagram above, we think n = 300 is large enough in interval 1.

(2) How large n is needed for bootstrap percentile method interval?

n = 30 is large enough for bootstrap percentile method interval.

From the diagrams in (b) and (c), we can see that the function converges at n = 30 in interval 2.

(3) Do these answers depend on $\boldsymbol{\lambda}$?

No. From the first four diagrams we got in (b), The change on $\boldsymbol{\lambda}$ does not affect the result. Also, thinking from mathematics way, the change on $\boldsymbol{\lambda}$ does not change the overall shape of exponential distribution. Hence, the relationship between the mean and the confidence interval remains. The accuracy would not change by $\boldsymbol{\lambda}$.

(4) Can we say that one method is more accurate than the other?

Yes, interval 2 is better than interval 1. From all eight diagrams, the accuracy of interval 2 is always larger than interval 1.

(5) Which interval would you recommend?

I would recommend interval 2 if we are talking about the accuracy of the confidence interval, especially for small n. However, if n is large enough, for example n = 100, calculation for interval 1 is quicker than interval 2 while the accuracy is close.

Overall, the bootstrap is good for estimating when the sample size is not large enough. However, it takes more resource to calculate while extra sampling in bootstrap. This can be seen from question (1) and (2).

(d) Do your conclusion in (c) depend on $\boldsymbol{\lambda}$ that were fixed in advance?

Actually, question (d) is quite similar the question (c) (3).

For different $\boldsymbol{\lambda}$, the accuracy does not change. Hence:

Answer for (1) remains: The n = 300 is large enough for large-sample z-interval.

Answer for (2) remains: The n = 30 is large enough for bootstrap percentile method interval.

Answer for (3) remains: The change on $\boldsymbol{\lambda}$ does not affect the relationship between the mean and the confidence interval. All answers do not depend on $\boldsymbol{\lambda}$.

Answer for (4) remains: The interval 2 is always more accurate. For its accuracy is always larger than interval 1's.

Answer for (5) remains: Still recommend interval 2. Only exceptional for large n on the consideration of calculation efficiency.