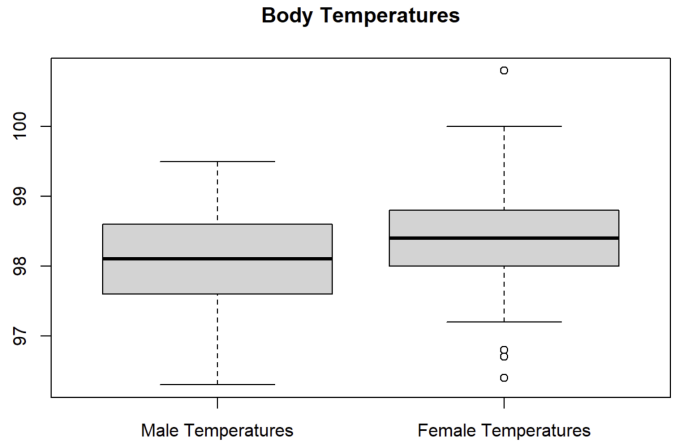
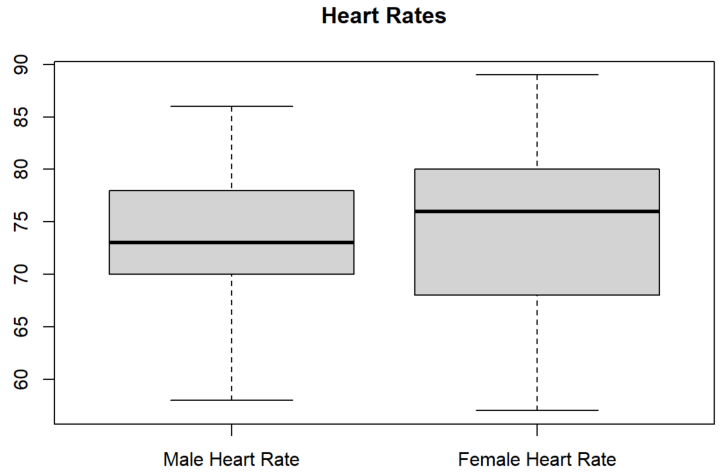
Mini Project #4
David McCormick (DTM190000)
I am in a solo group.

Section 1

1. Part 1
    a. The mean body temperature for males is 98.1 degrees Fahrenheit whereas the mean body temperature for females is 98.4 degrees Fahrenheit, representing a slight difference between the two means. According to the boxplot below, the female temperatures have a slightly higher median and more outliers than the males. They do not appear to have equal variances.

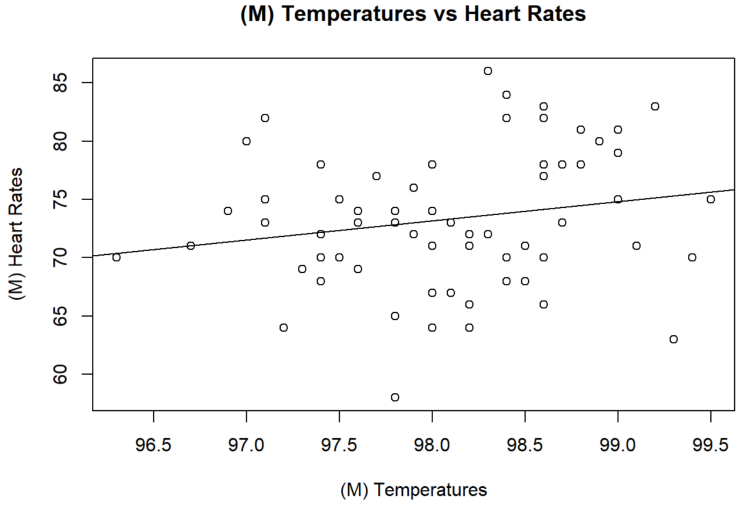


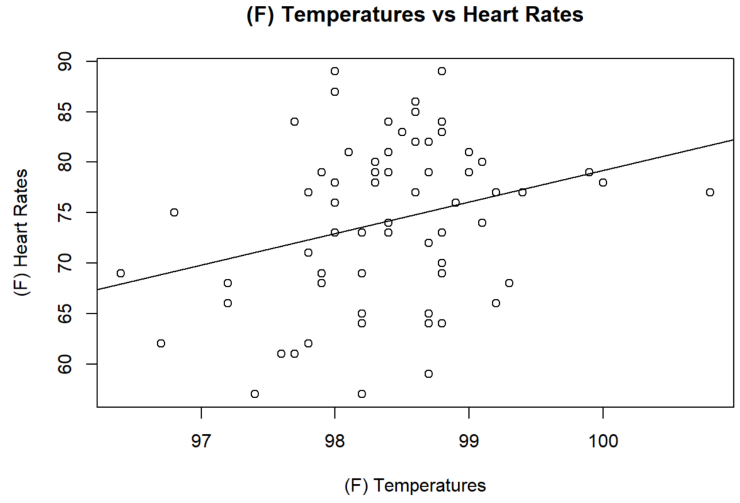
    b. The mean heart rate for males is 73.4 beats per minute (bpm) whereas the mean heart rate for females is 74.15 bpm., representing a difference of about 1 beat per minute. According to the boxplot below, male heart rate has a lower median than female heart rate. Additionally, the variance of female heart rate is much greater than male heart rate.



c. According to the scatterplots below, there seems to be a linear relationship between body temperature and heart rate. Both of the lines have positive slopes, representing a slight positive relationship. As male and female temperature increase, so do their heart rates.

   In addition, we can calculate the correlation between the male temperatures and heart rates as well as for females. Male correlation has a value of .196, whereas female correlation has a value of .287. Thus, we can deduce that the strength of the relationship is slightly stronger for females than males.





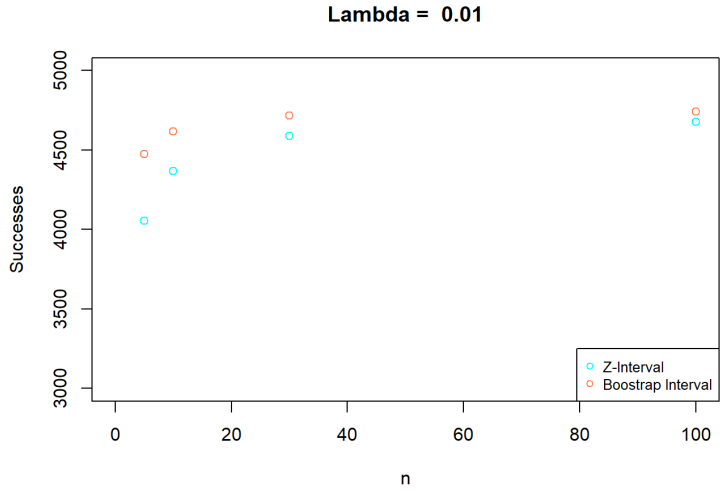
2. Part 2
   a. For this part, I calculated the coverage probabilities for the Z-interval and the percentile bootstrap method with an alpha of .05, lambda of .01, and n of 5. The large sample Z-interval produced 4054/5000 (81.08%) correct intervals, whereas the percentile bootstrap method produced 4476/5000 (89.52%) correct intervals.

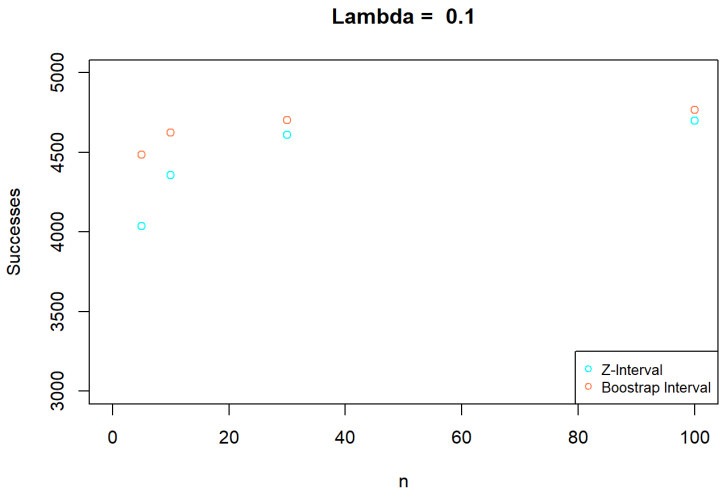
b. After repeating the process for all values of lambda and n, the resulting tables are below:

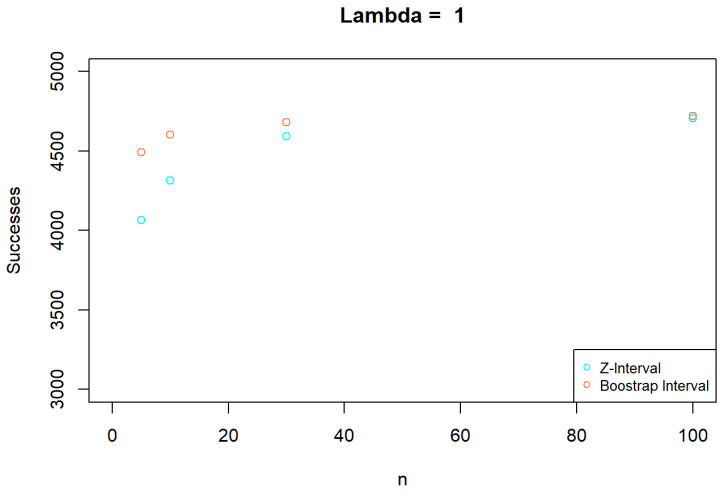
Z-Interval	$\lambda = .01$	$\lambda = .1$	$\lambda = 1$	$\lambda = 10$
$n = 5$	4054	4035	4064	4059
$n = 10$	4366	4357	4314	4332
$n = 30$	4589	4609	4591	4592
$n = 100$	4678	4698	4705	4673

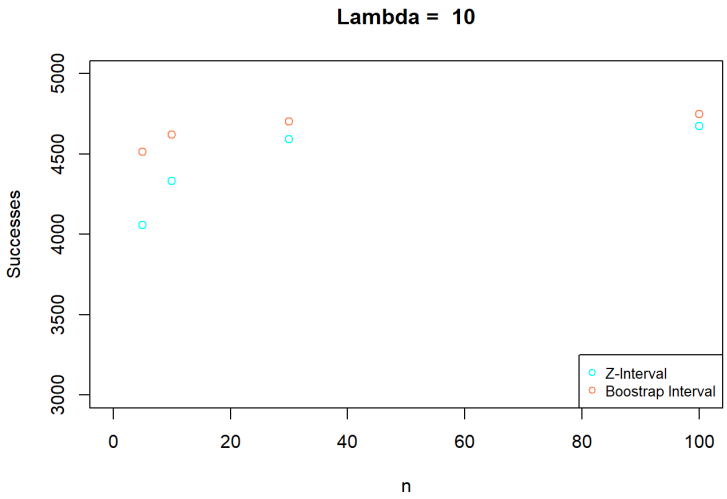
Bootstrap	$\lambda = .01$	$\lambda = .1$	$\lambda = 1$	$\lambda = 10$
$n = 5$	4476	4486	4493	4513
$n = 10$	4617	4624	4602	4619
$n = 30$	4718	4702	4680	4702
$n = 100$	4742	4768	4720	4750

Below are the results graphically represented, organized by lambda value:

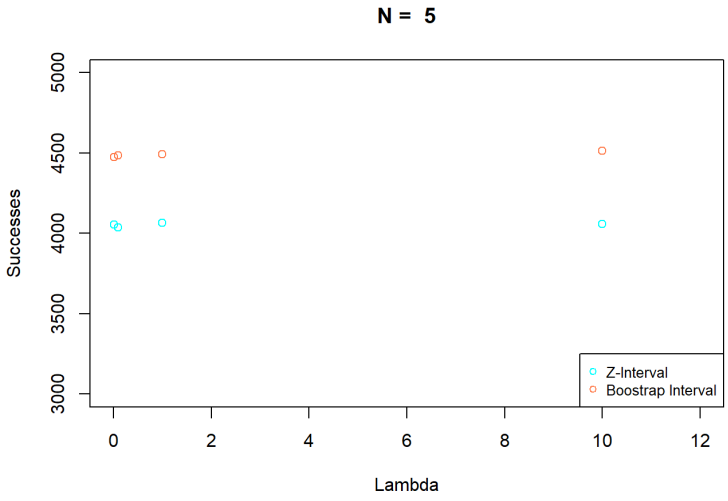


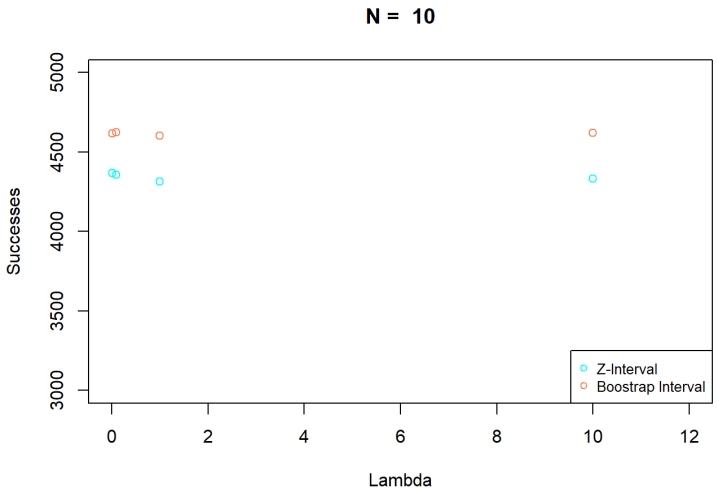

Lambda = 0.1
Successes
n
Z-Interval
Boostrap Interval
3000
3500
4000
4500
5000
0
20
40
60
80
100

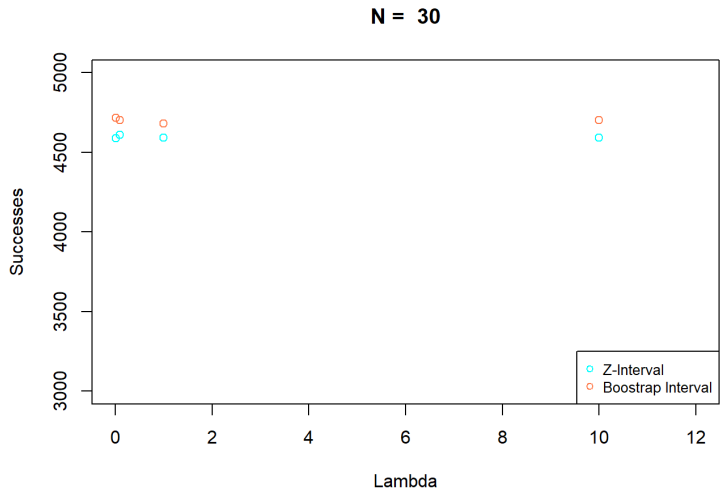

Lambda = 1
Successes
n
Z-Interval
Boostrap Interval
3000
3500
4000
4500
5000
0
20
40
60
80
100

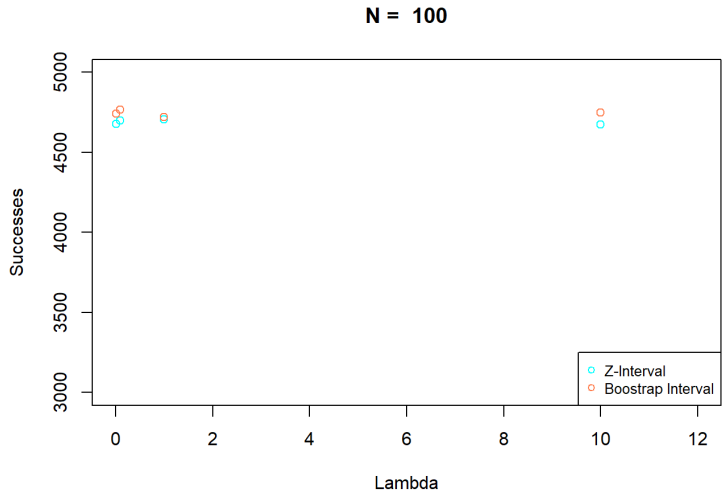

Lambda = 10
Successes
n
Z-Interval
Boostrap Interval
3000
3500
4000
4500
5000
0
20
40
60
80
100

Finally, below are the results graphically represented, organized by n:









c. After reviewing the results, we can conclude that larger sample sizes lead to greater amounts of success for both bootstrapping methods. On average, the bootstrap method was performing better for each pairing of sample size and lambda. The bootstrap method achieves a level of 90% success rate for n = 5, and tapers out at around a level of 95% success rate for n = 100. On the other hand, the Z-interval method achieves a level of about 80% success rate for n = 5 and 93% percent success rate for n = 100.

I would argue that a sample size of n = 30 is sufficient to provide accurate results for the bootstrapping method, whereas the Z-interval should be provided with n = 100 to provide accurate results.

Additionally, we can determine that the results do not depend on lambda. Rather, they depend on the sample size, which can be determined by noting that the

graphs sorted by lambda value are similar whereas the graphs sorted by n are different.

Overall, I would recommend the bootstrapping method because for the same amount of sample size, it provides more accurate results on average. I would note, however, that the bootstrapping method seemed more computationally intensive than the Z-interval method because it took longer to compute.

d. No, the conclusions in part C look at the data holistically. The bootstrapping method outperformed the Z-interval method at every pairing of lambda and n.

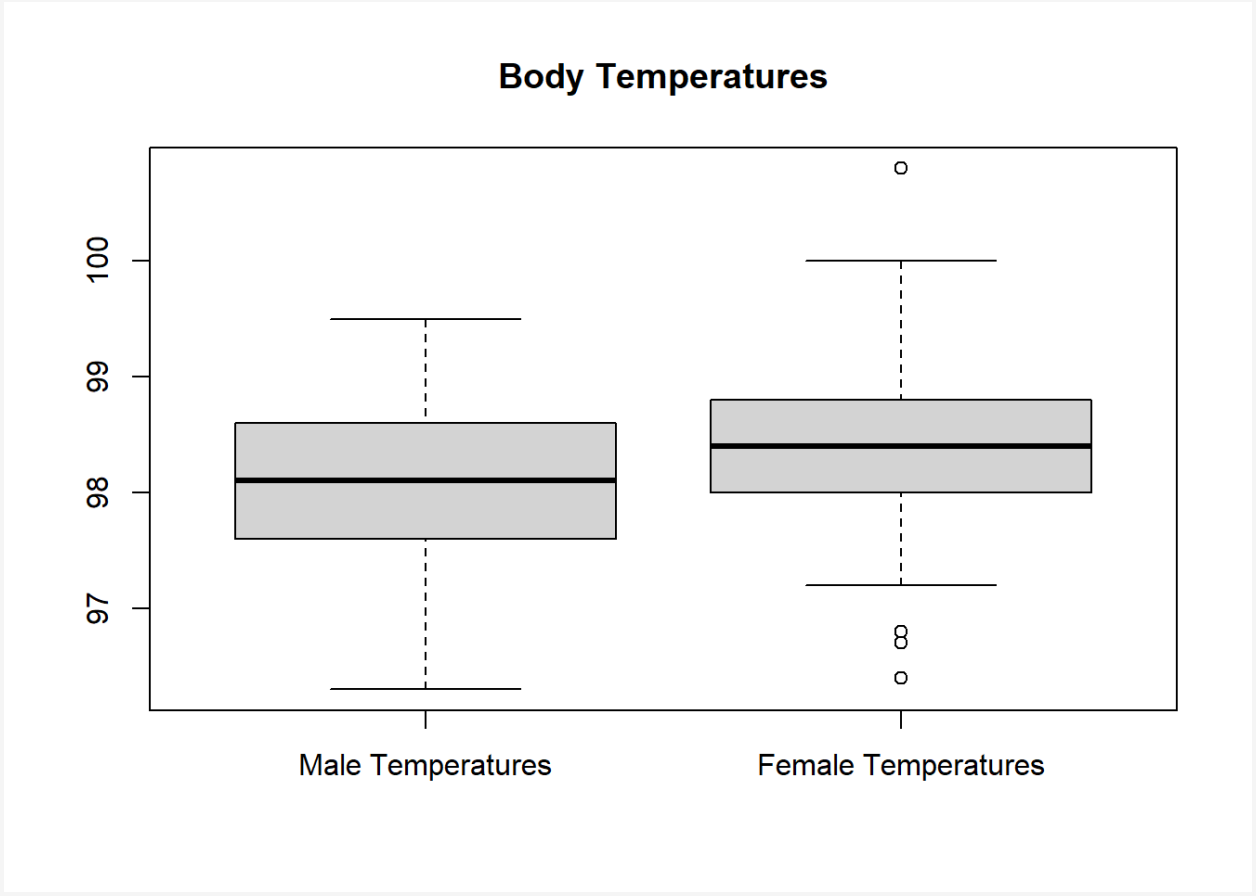
Part 2

project5.R

David

2021-11-19

```
#### Question 1 ####


# Reads the csv file
data = read.csv("C:/Users/David/Desktop/bodytemp-heartrate.csv")


# Provides basic statistics
summary <- function(col) {
  print(mean(col))
  print(median(col))
  print(var(col))
}


# Separates the data into temperature and heart rate
temp = data[,c(1)]
heart = data[,c(3)]


# Separates the data into female and male
mtemp = data$body_temperature[which(data$gender == 1)]
ftemp = data$body_temperature[which(data$gender == 2)]
mheart = data$heart_rate[which(data$gender == 1)]
fheart = data$heart_rate[which(data$gender == 2)]


# Boxplots to compare distributions for temperature
boxplot(mtemp, ftemp, names=c("Male Temperatures", "Female Temperatures"), main="Body
Temperatures")
```




```
summary(mtemp)
```

[1] 98.10462

[1] 98.1

[1] 0.4882596

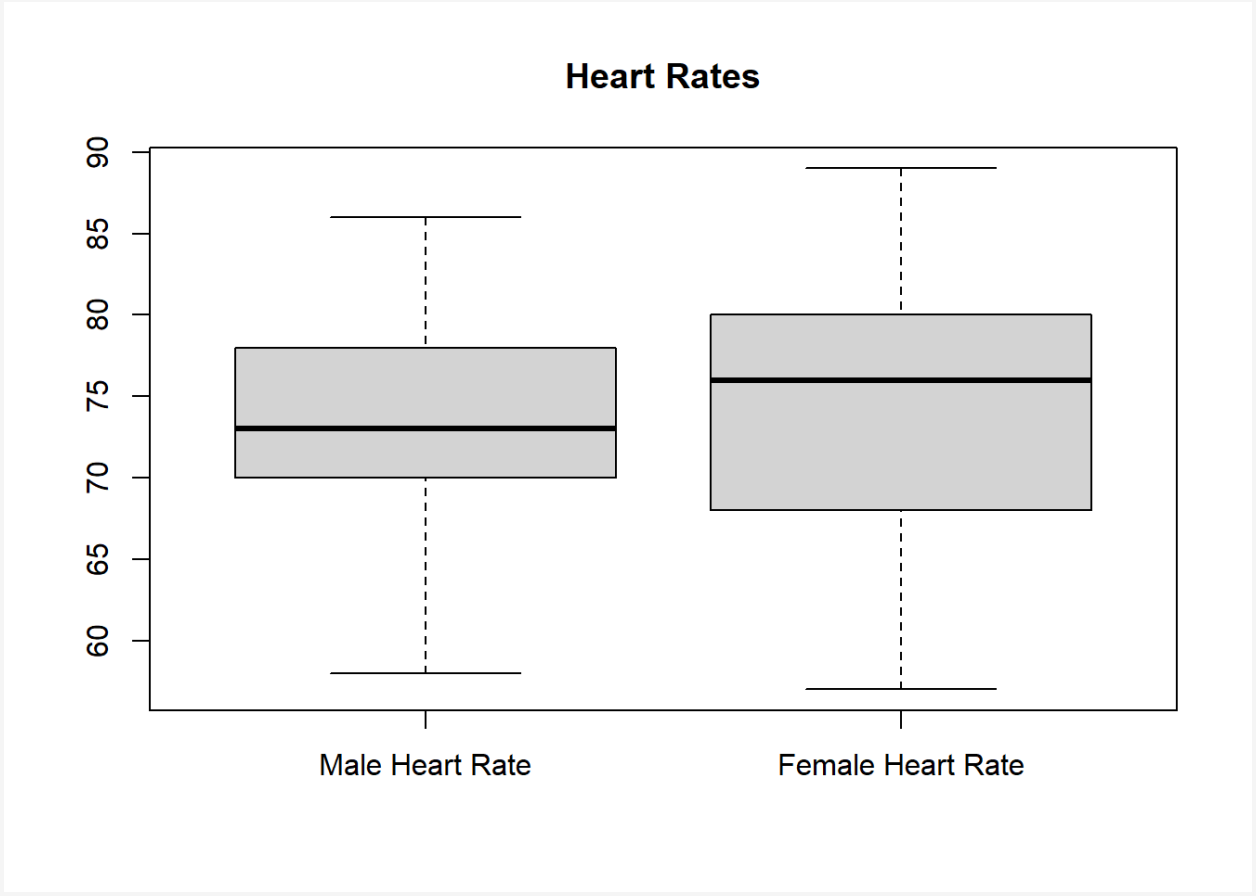
```
summary(ftemp)
```

[1] 98.39385

[1] 98.4

[1] 0.552774

```
# Boxplots to compare distributions for heart rate
boxplot(mheart, fheart, names=c("Male Heart Rate", "Female Heart Rate"), main="Heart Rates")
```




```
summary(mheart)
```

```
## [1] 73.36923
```

```
## [1] 73
```

```
## [1] 34.51779
```

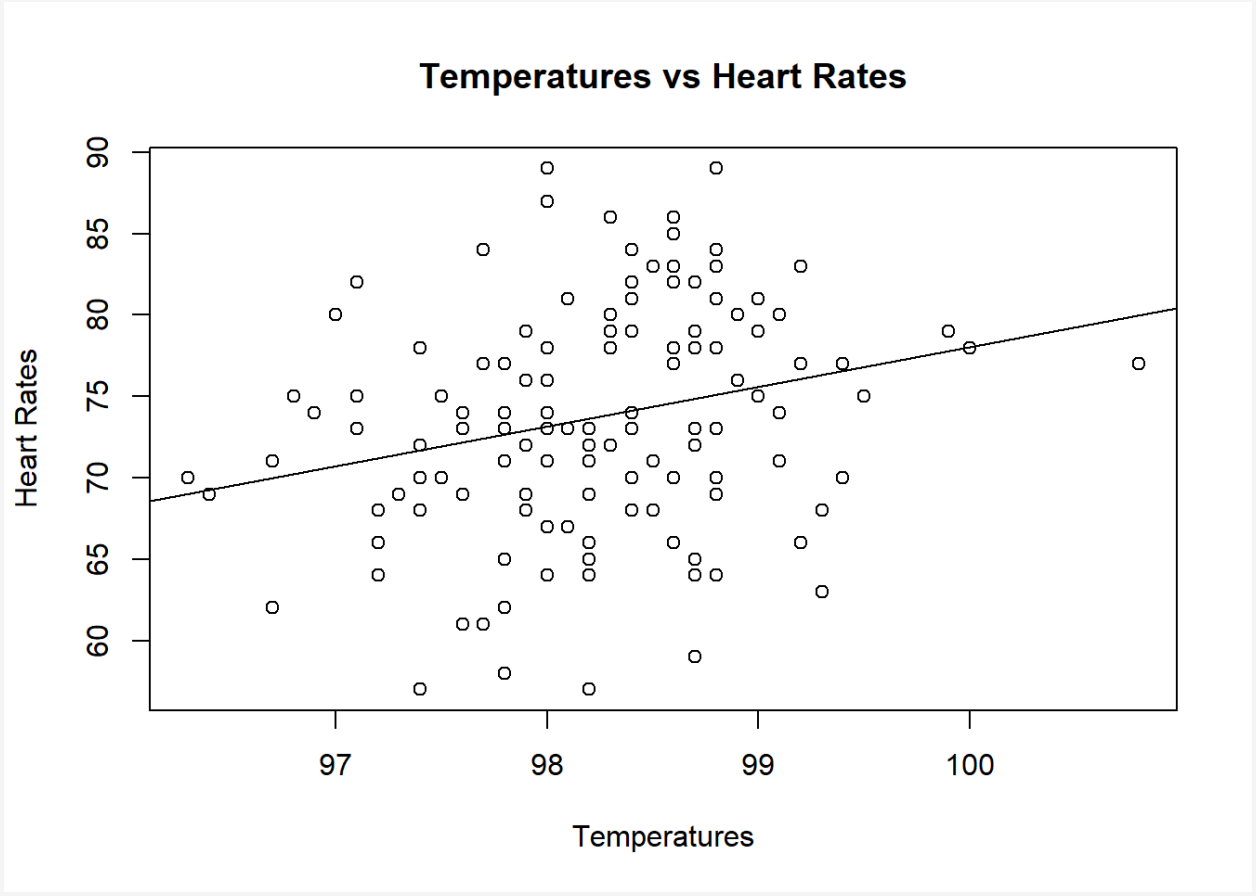
```
summary(fheart)
```

```
## [1] 74.15385
```

```
## [1] 76
```

```
## [1] 65.69471
```

```
# Scatterplot for male and female temperature and heart rate
plot(temp, heart, xlab="Temperatures", ylab="Heart Rates", main="Temperatures vs Heart Rates")
abline(lm(heart~temp))
```



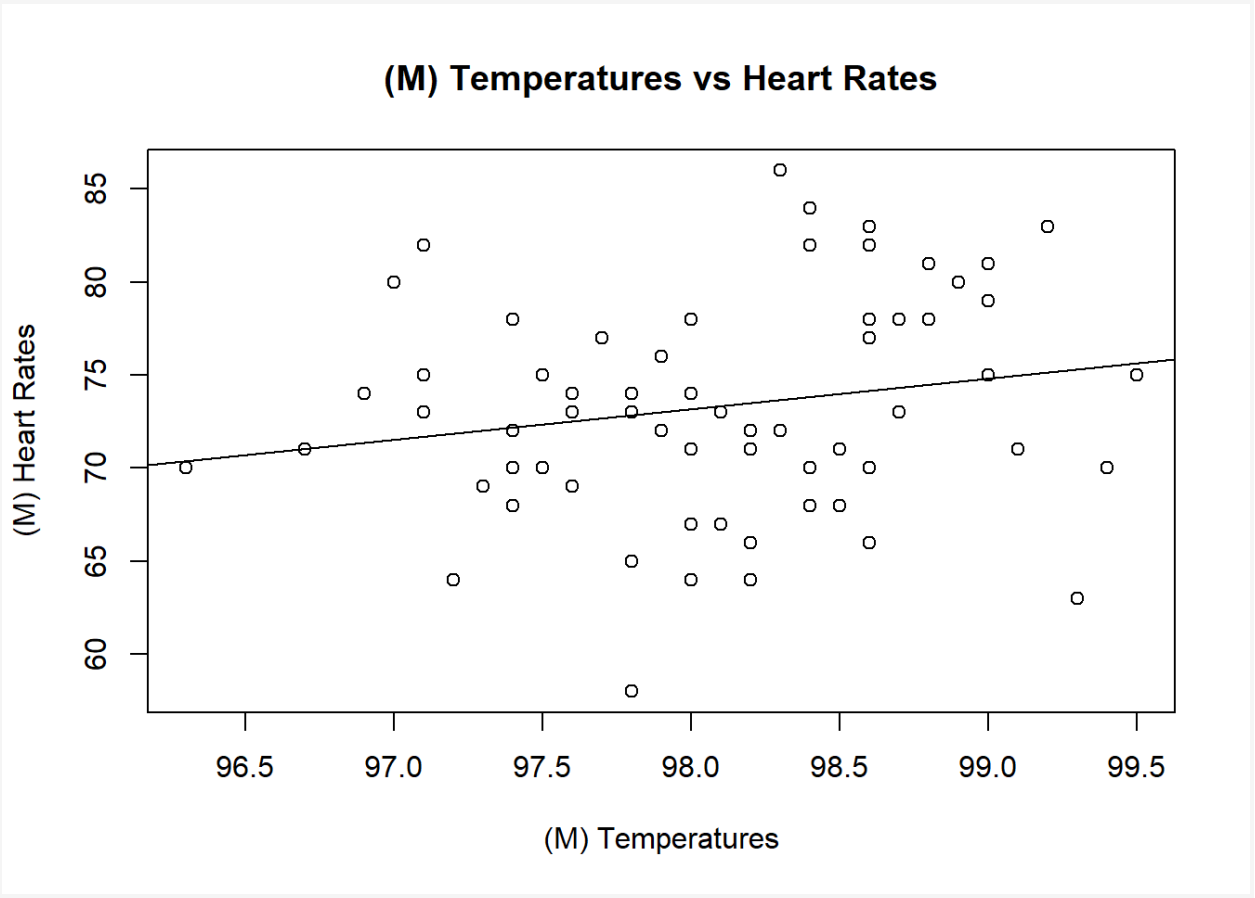

```
cor(temp, heart)
```

```
## [1] 0.2536564
```

```
# Scatterplot for male temperature and heart rate
```

```
plot(mtemp, mheart, xlab="(M) Temperatures", ylab="(M) Heart Rates", main="(M) Temperatures vs Heart Rates")
```

```
abline(lm(mheart~mtemp))
```



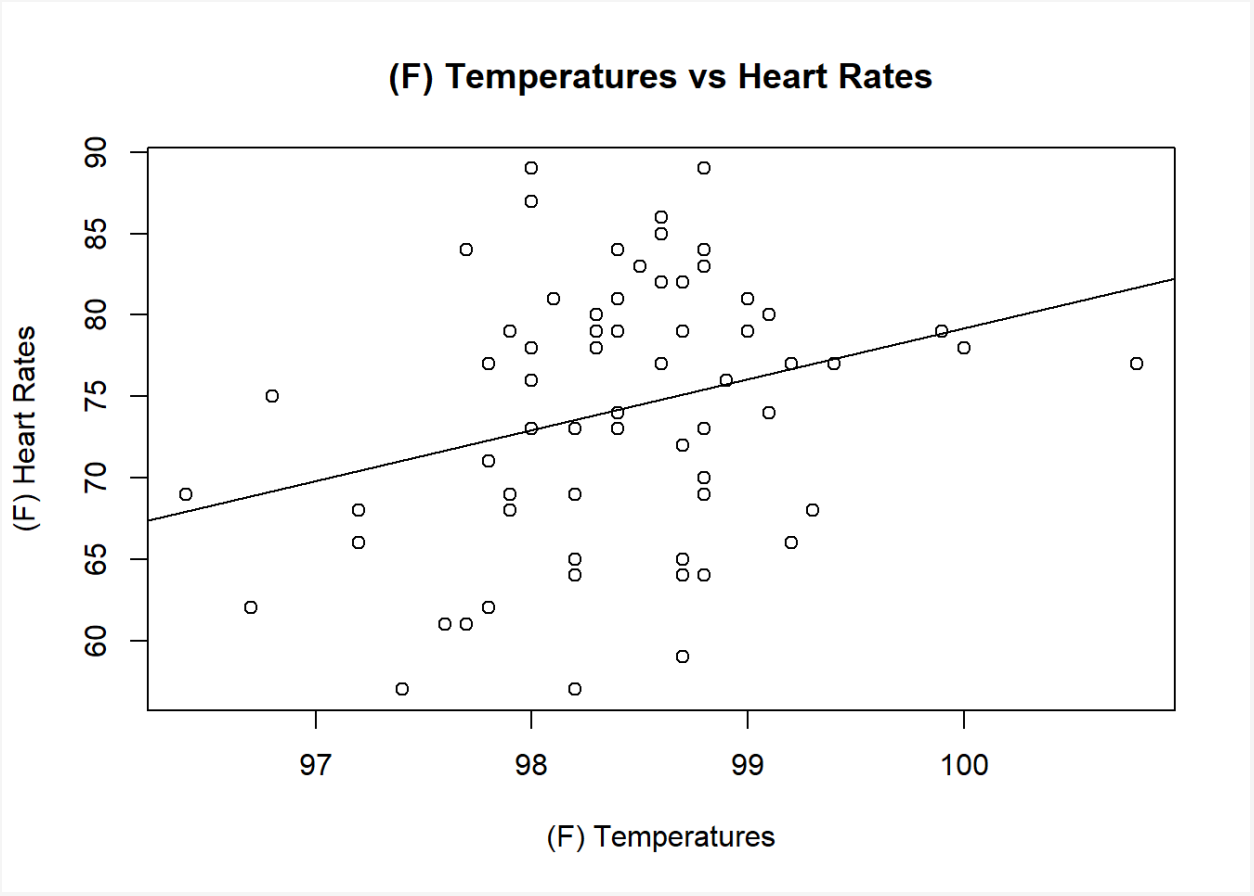

```
cor(mtemp, mheart)
```

```
## [1] 0.1955894
```

```
# Scatterplot for female temperature and heart rate
```

```
plot(ftemp, fheart, xlab="(F) Temperatures", ylab="(F) Heart Rates", main="(F) Temperatures vs Heart Rates")
```

```
abline(lm(fheart~ftemp))
```




```
cor(ftemp, fheart)
```

```
## [1] 0.2869312
```

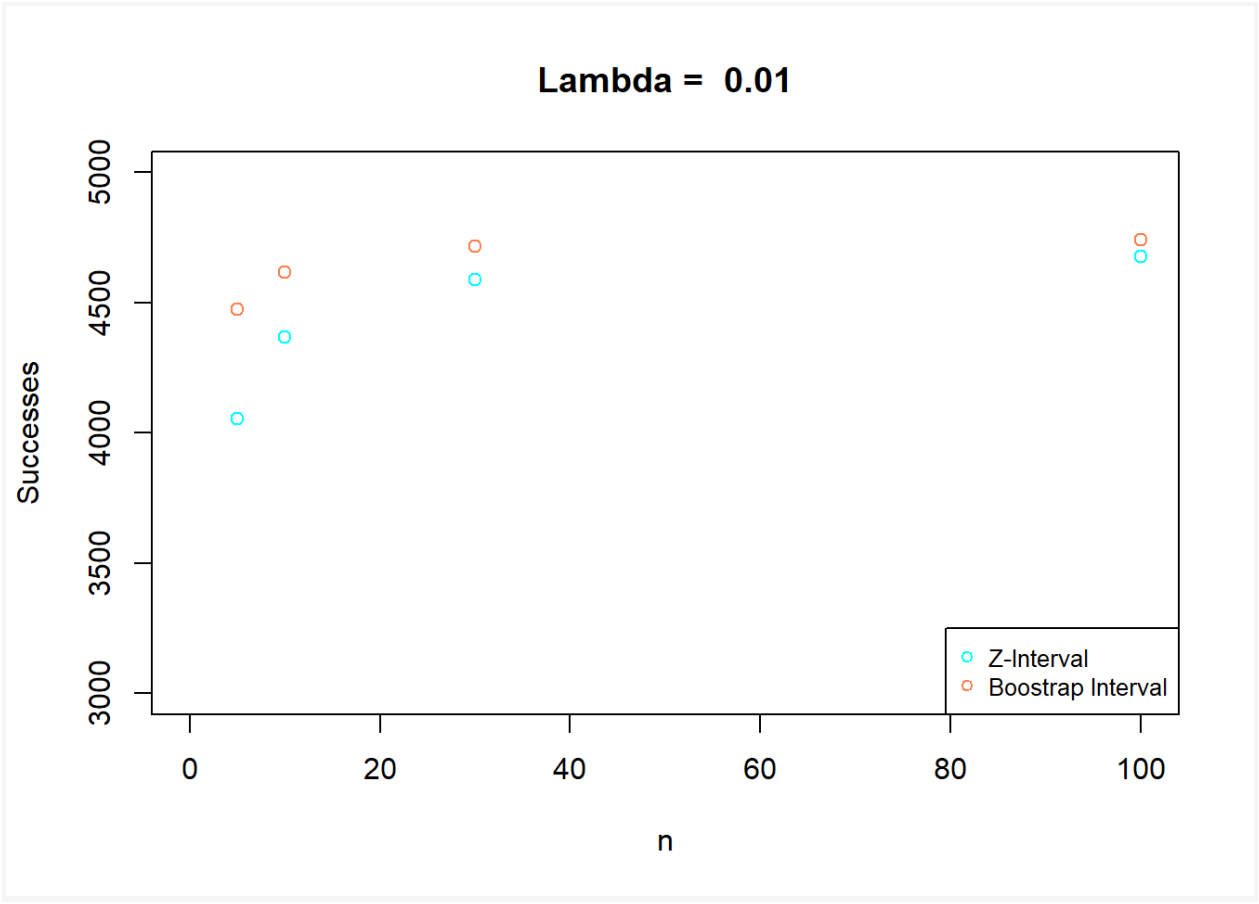
```
#### Question 2 ####
```

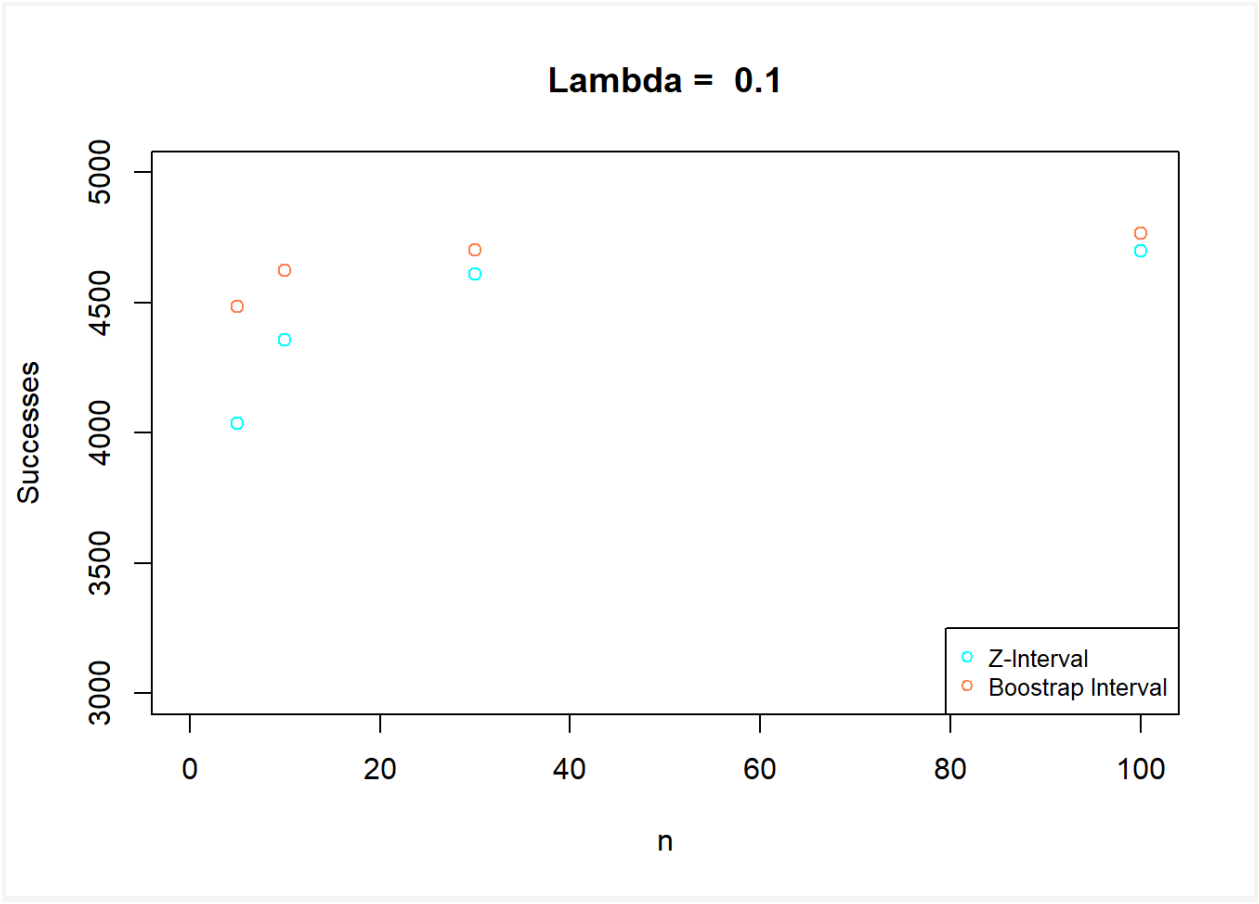
```
n_arr = c(5, 10, 30, 100)
lambda_arr = c(.01, .1, 1, 10)
alpha = .05

# Creates a Z-interval and sees if the true value (1/lambda) is captured by the
#  interval. Returns 1 if the true value is captured and 0 if the true value is
#  not captured
checkZInt <- function(n, lambda) {
  distr = rexp(n, rate=lambda)
  se = sd(distr)/sqrt(n)
```

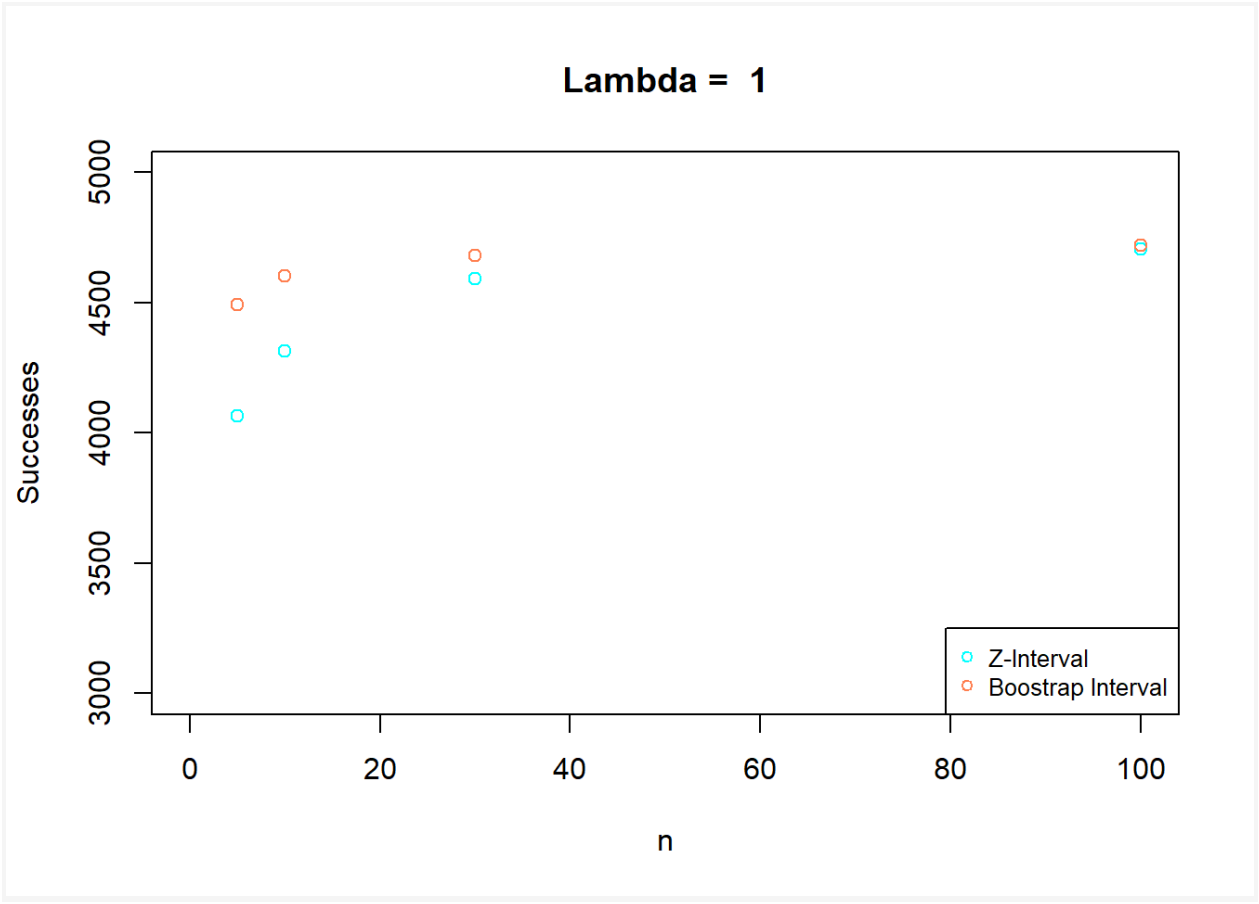
```
  upper = mean(distr) + qnorm(1-.025) * se
  lower = mean(distr) - qnorm(1-.025) * se
  tm = 1/lambda
  if (lower < tm && tm < upper) {
    return (1)
  }
  return (0)
}

# Performs the n/lambda combination 5000 times and returns how many successes
#  there were
zTrials <- function(n, lambda) {
  values = replicate(5000, checkZInt(n, lambda))
  success = values[which(values==1)]
  return (length(success))
}

# For bootstrapping
mean_ <- function(n, lambda) {
  distr = rexp(n, lambda)
  return (mean(distr))
}

# Creates a percentile bootstrap interval and sees if the true value (1/lambda)
#  is captured by the interval. Returns 1 if the true value is captured and 0
#  if the true value is not captured. The confidence interval is 95%, thus taking
#  the 25th and 975th sorted values will provide the confidence interval.
checkBInt <- function(n, lambda) {
  distr = rexp(n, lambda)
  tm = 1/lambda
```

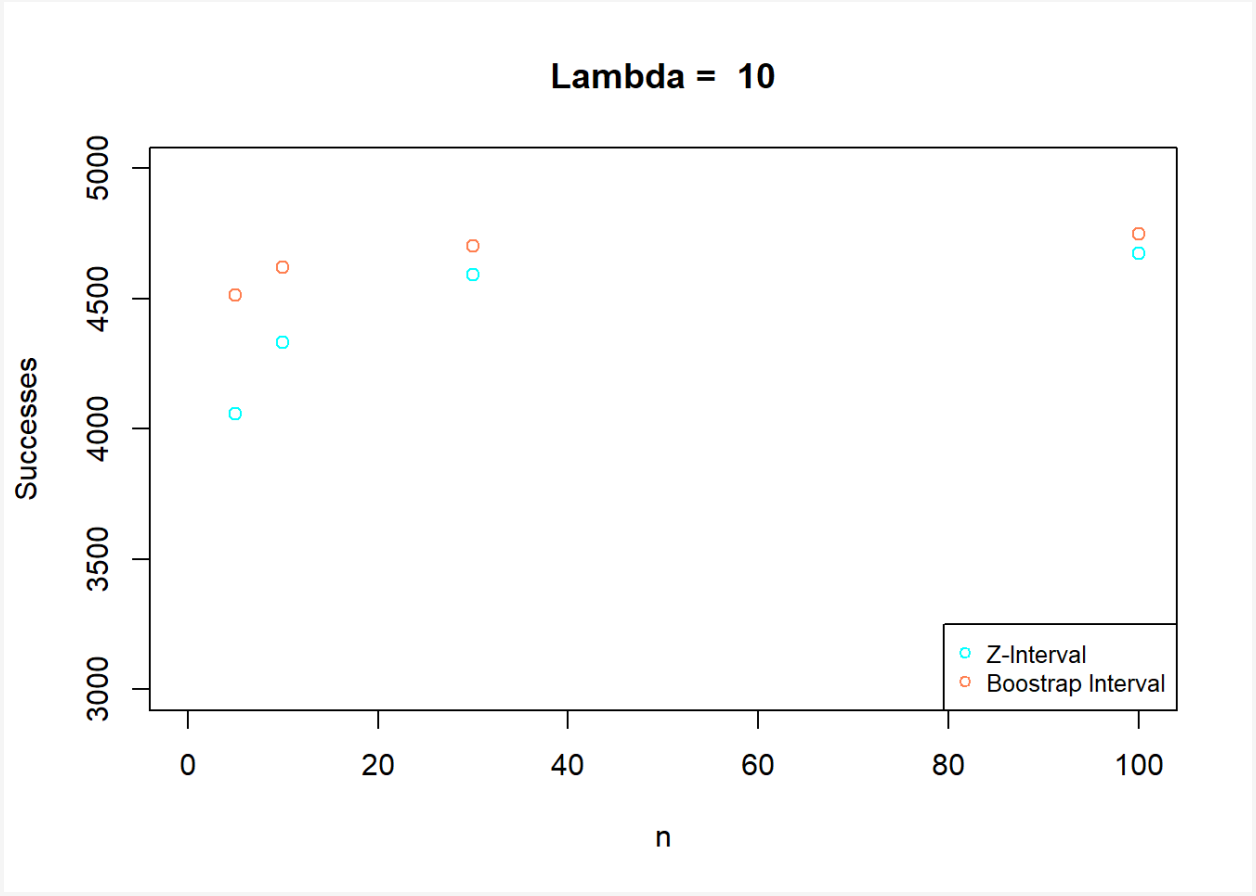
```
  lambda1 = 1/(mean(distr))
  results = replicate(1000, mean_(n, lambda1))
  lower = sort(results)[25]
  upper = sort(results)[975]
  if (lower < tm && tm < upper) {
    return (1)
  }
  return (0)
}

# Performs the n/lambda combination 5000 times and returns how many successes
#  there were
bTrials <- function(n, lambda) {
  values = replicate(5000, checkBInt(n, lambda))
  success = values[which(values==1)]
  return (length(success))
}

zRes = c()
bRes = c()
# THIS IS WHERE THE ACTUAL CALCULATIONS ARE PERFORMED, UNCOMMENT THIS TO RUN
THEM
#for (lambda in lambda_arr){ # Performs all of the trials by iterating through arrays
#  for (n in n_arr){
#    zRes <- rbind(zRes, zTrials(n, lambda))
#    bRes <- rbind(bRes, bTrials(n, lambda))
#  }
#}

# I MOVED THE RESULTS HERE, SO I WOULDN'T HAVE TO RUN THE CALCULATIONS EVERYTIME
# SEE THE ABOVE BLOCK OF CODE TO SEE HOW THESE ARE CALCULATED
```

```
# zRes = z-interval results sorted by N (sample size) or L (lambda)

# bRes = percentile bootstrap results sorted, again, by N (sample size) or L (lambda)

zResN = c(4054,4366,4589,4678,4035,4357,4609,4698,4064,4314,4591,4705,4059,4332,4592,4673)

zResL = c(4054,4035,4064,4059,4366,4357,4314,4332,4589,4609,4591,4592,4678,4698,4705,4673)

bResN = c(4476,4617,4718,4742,4486,4624,4702,4768,4493,4602,4680,4720,4513,4619,4702,4750)

bResL = c(4476,4486,4493,4513,4617,4624,4602,4619,4718,4702,4680,4702,4742,4768,4720,4750)


# Graphs the results sorted by lambda

for (iter in seq(1,length(lambda_arr))) {

  plot(n_arr[(1):(4)], zResN[(4 * iter-3):(4 * iter)], xlim=c(0,100), ylim=c(3000,5000), xlab="n",
ylab="Successes", main=paste("Lambda = ", lambda_arr[iter]), col="cyan")

  points(n_arr[(1):(4)], bResN[(4 * iter-3): (4 * iter)], col="coral")

  legend("bottomright", legend=c("Z-Interval", "Boostrap Interval"), col=c("cyan", "coral"), pch=1:1,cex=0.8)

}
```

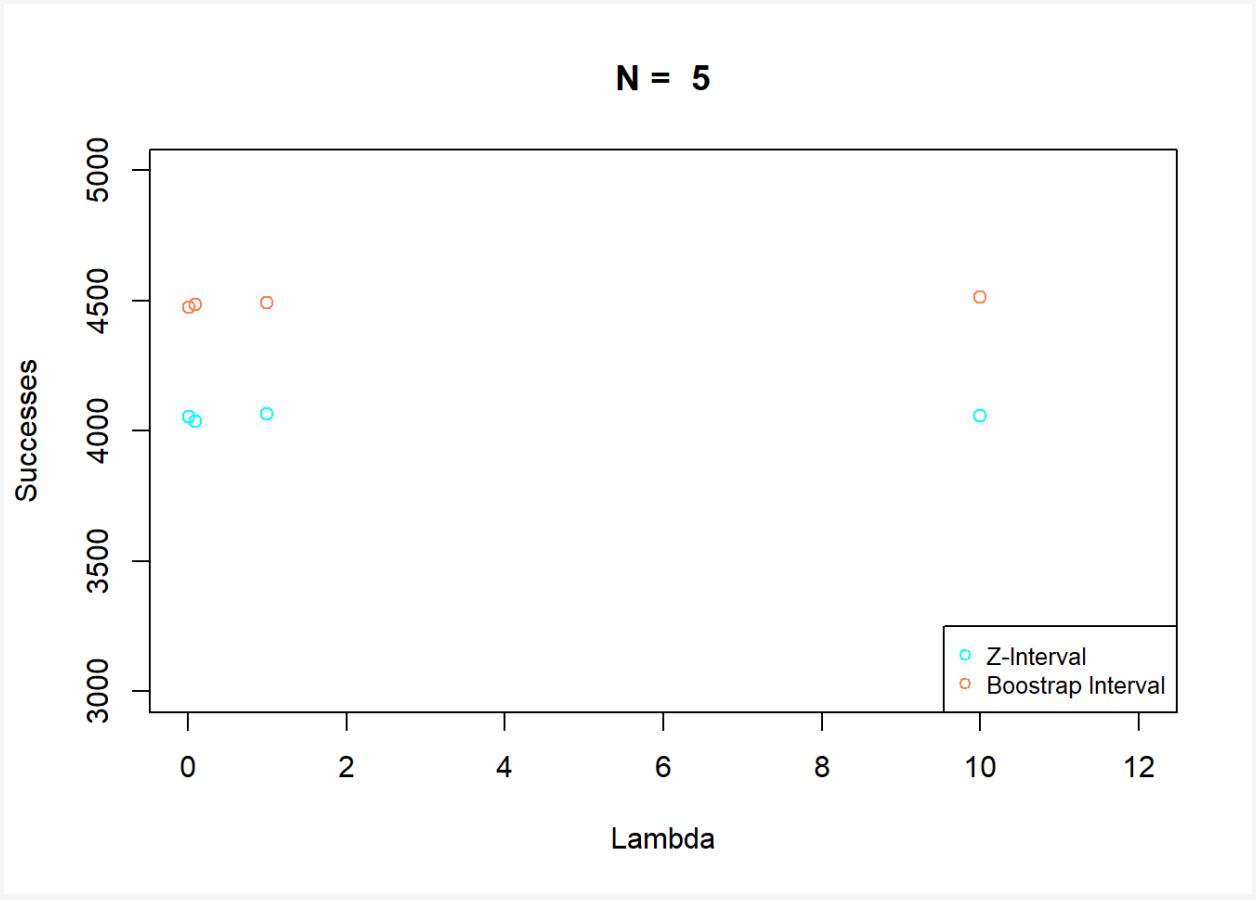

Lambda = 0.01
Successes
3000
3500
4000
4500
5000
0
20
40
60
80
100
n
Z-Interval
Boostrap Interval

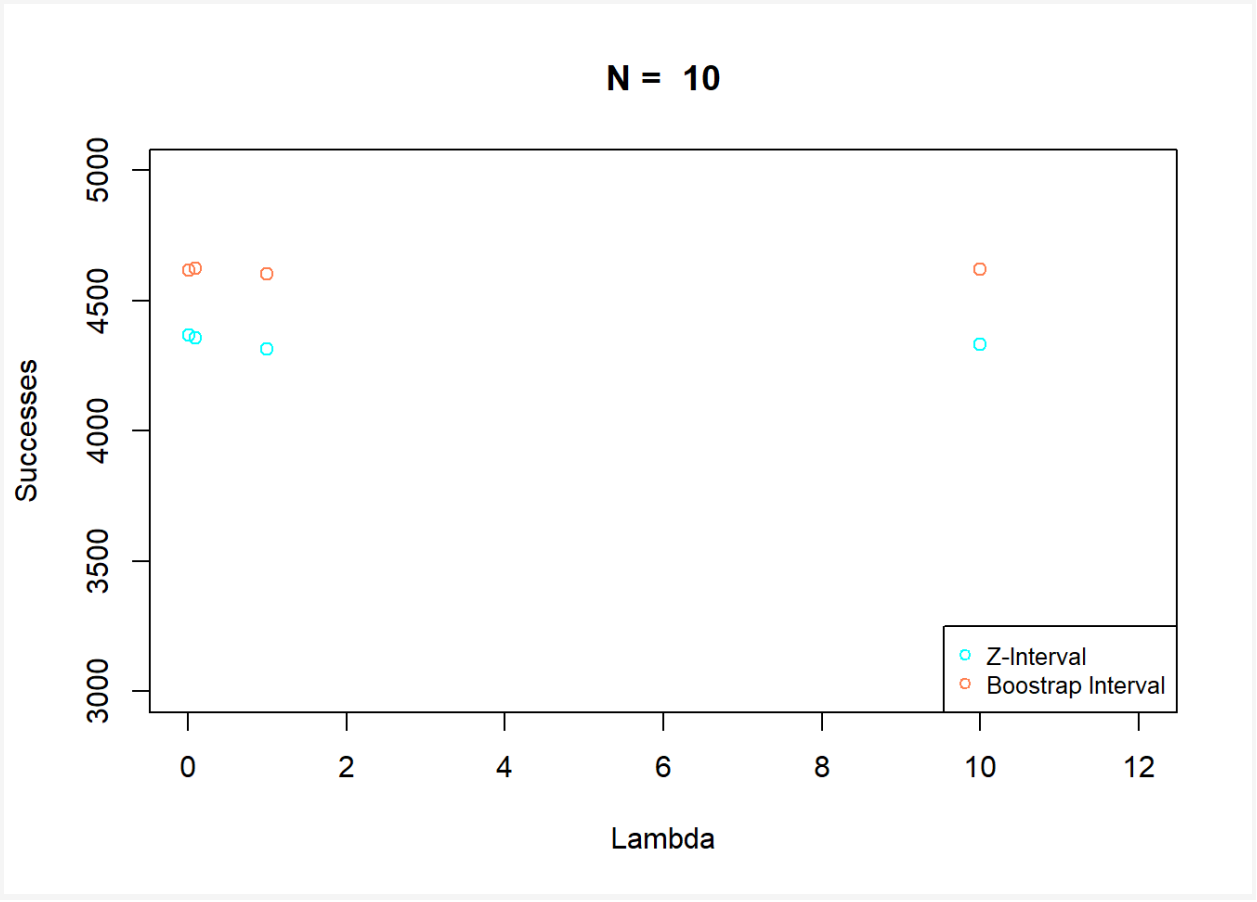

Lambda = 0.1
Successes
3000
3500
4000
4500
5000
0
20
40
60
80
100
n
Z-Interval
Boostrap Interval

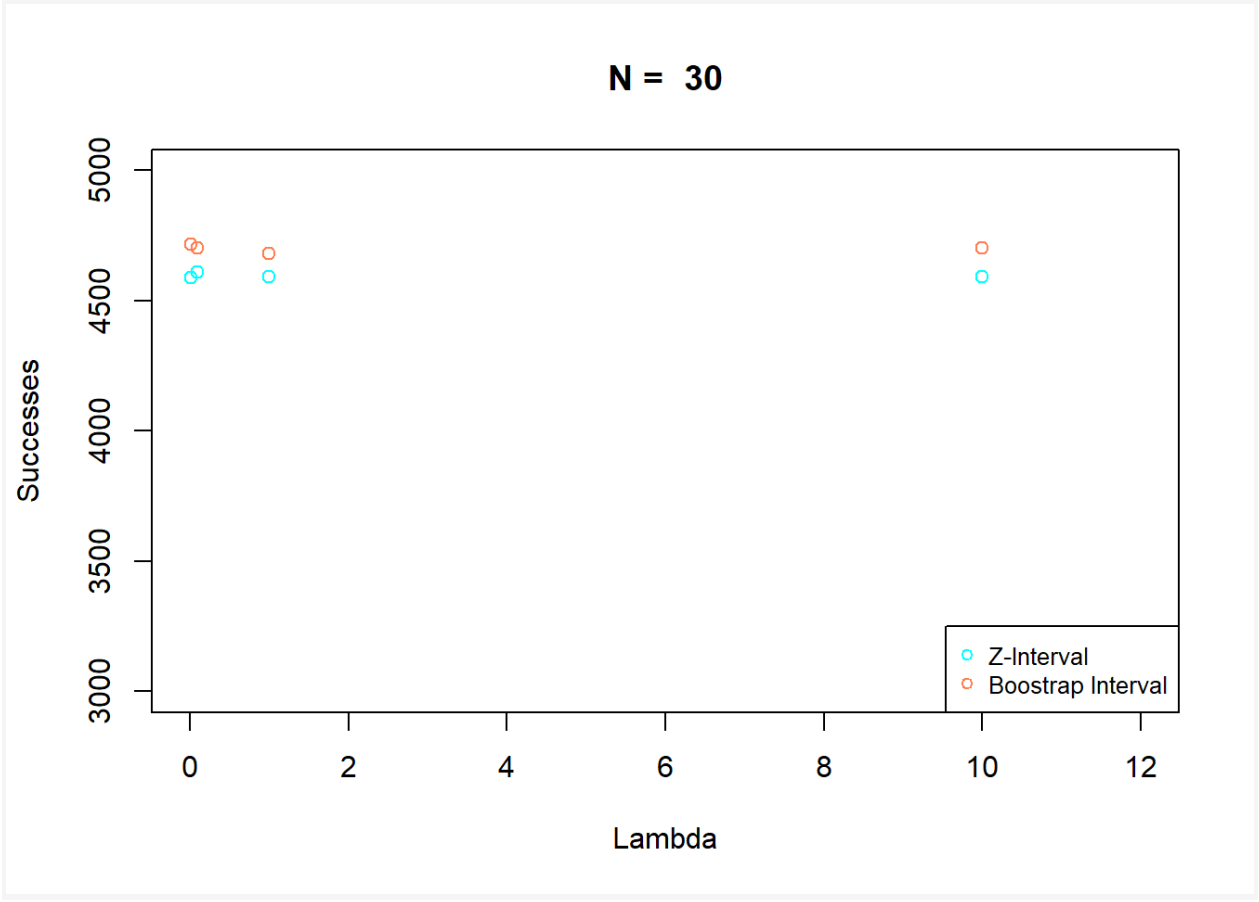

Lambda = 1
Successes
3000
3500
4000
4500
5000
0
20
40
60
80
100
n
Z-Interval
Boostrap Interval

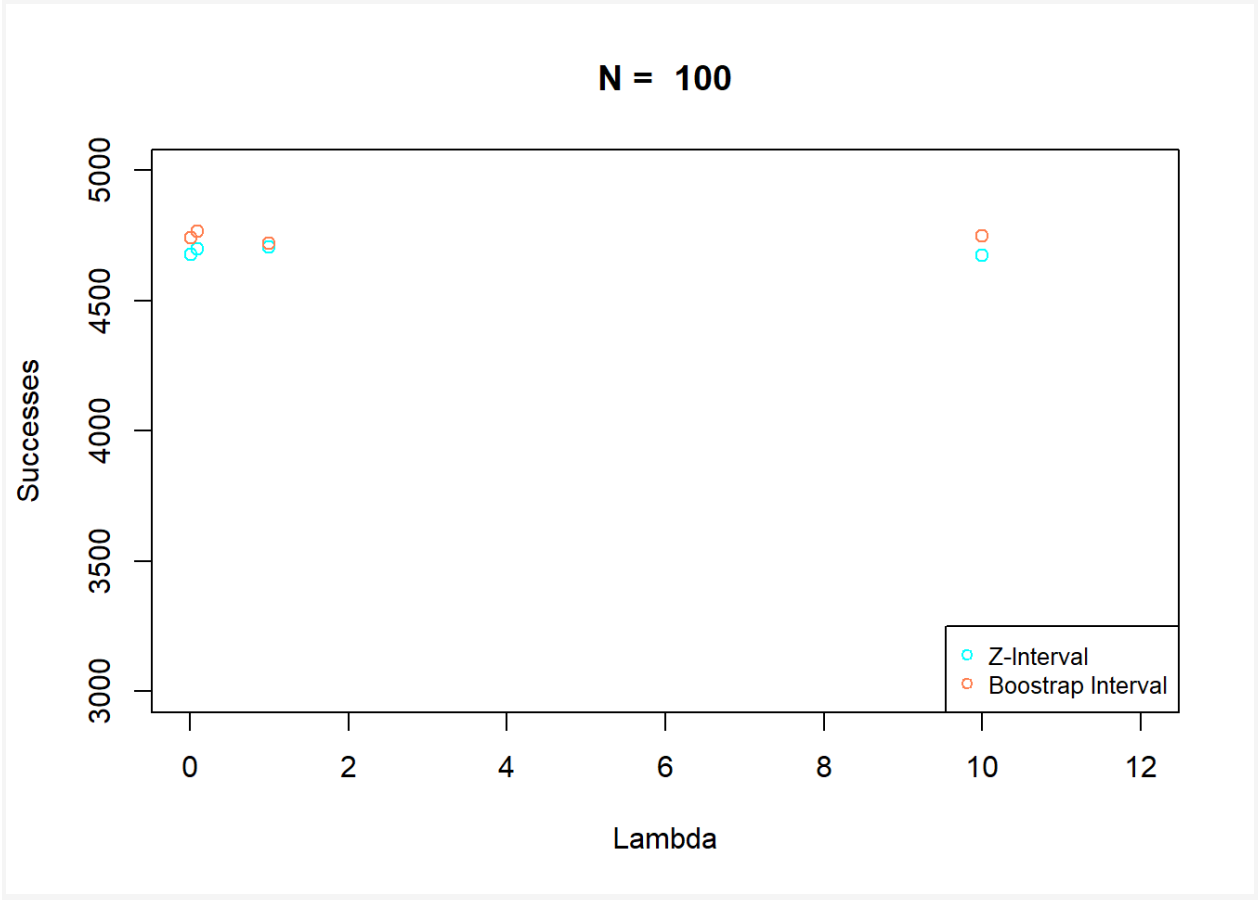



```
# Graphs the results sorted by n

for (iter in seq(1,length(n_arr))) {

  plot(lambda_arr[(1):(4)], zResL[(4 * iter-3):(4 * iter)], xlim=c(0,12), ylim=c(3000,5000), xlab="Lambda",
ylab="Successes", main=paste("N = ", n_arr[iter]), col="cyan")

  points(lambda_arr[(1):(4)], bResL[(4 * iter-3): (4 * iter)], col="coral")

  legend("bottomright", legend=c("Z-Interval", "Boostrap Interval"), col=c("cyan", "coral"), pch=1:1,cex=0.8)

}
```


N = 5
Successes
3000
3500
4000
4500
5000
0
2
4
6
8
10
12
Lambda
Z-Interval
Boostrap Interval


N = 10
Successes
5000
4500
4000
3500
3000
0
2
4
6
8
10
12
Lambda
Z-Interval
Boostrap Interval


N = 30
Successes
3000
3500
4000
4500
5000
0
2
4
6
8
10
12
Lambda
Z-Interval
Boostrap Interval


N = 100
Successes
3000
3500
4000
4500
5000
0
2
4
6
8
10
12
Lambda
Z-Interval
Boostrap Interval