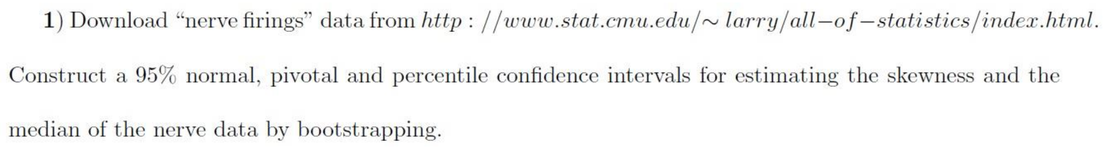
HW4_20201116_PeiShanYen

Question 01



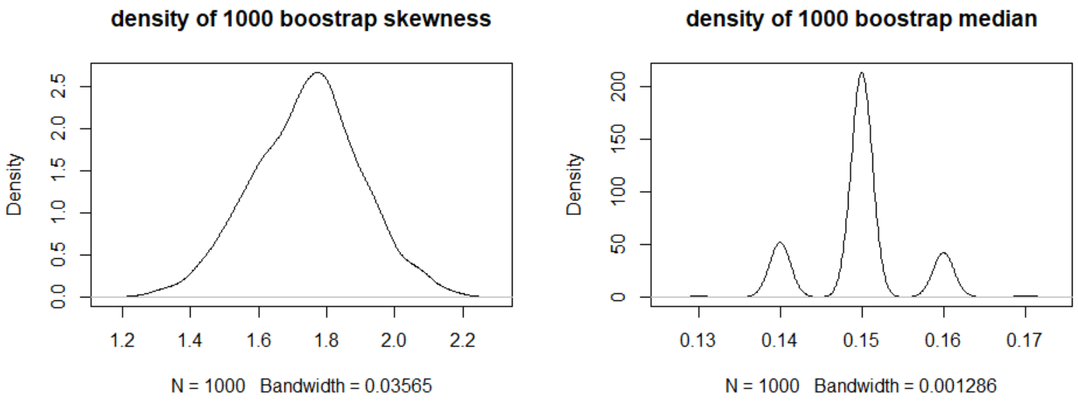
1) Download “nerve firings” data from http : //www.stat.cmu.edu/∼ larry/all−of−statistics/index.html.

Construct a 95% normal, pivotal and percentile confidence intervals for estimating the skewness and the median of the nerve data by bootstrapping.

Bootstrap resampling was used 1000 times to construct the estimation of the 95% confidence interval of 1) skewness and 2) median of the nerve data. The normal, pivotal, and percentile methods are performed to estimate the bootstrap confidence intervals. For this nerve dataset, the three methods produce similar results. The point estimate for skewness and the median is 1.761249 and 0.15. Their corresponding 95% percentile confidence interval are (1.432778, 2.060223) and (0.1400000, 0.1600000).

```
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

# Import Data
nerve = read.table("C:/Users/pyen2/Dropbox/UIC BSTT 565 Computational
Statistics/4.HW/HW4/nerve.dat", fill = T)
nerve = nerve %>% unlist %>% na.omit %>% as.numeric

# sample size
n = length(nerve)

# Function of Skewness
f.skew = function(x) {mean((x - mean(x))^3)/mean((x - mean(x))^2)^(3/2)}

# point estimation
skew.hat = f.skew(nerve)    # Skewness
skew.hat

## [1] 1.761249

median.hat = median(nerve)  # Median
median.hat

## [1] 0.15

# Bootstrap (Skewness, Median)
set.seed(20200824)
B = 1000
```

```
B.skew = B.median = numeric(length = B)
for (i in 1:B) {
  x = sample(nerve, size = n, replace = T)
  B.skew[i] = f.skew(x)
  B.median[i] = median(x)
}
rm(i, x)
```



95% bootstrap confidence interval for skewness

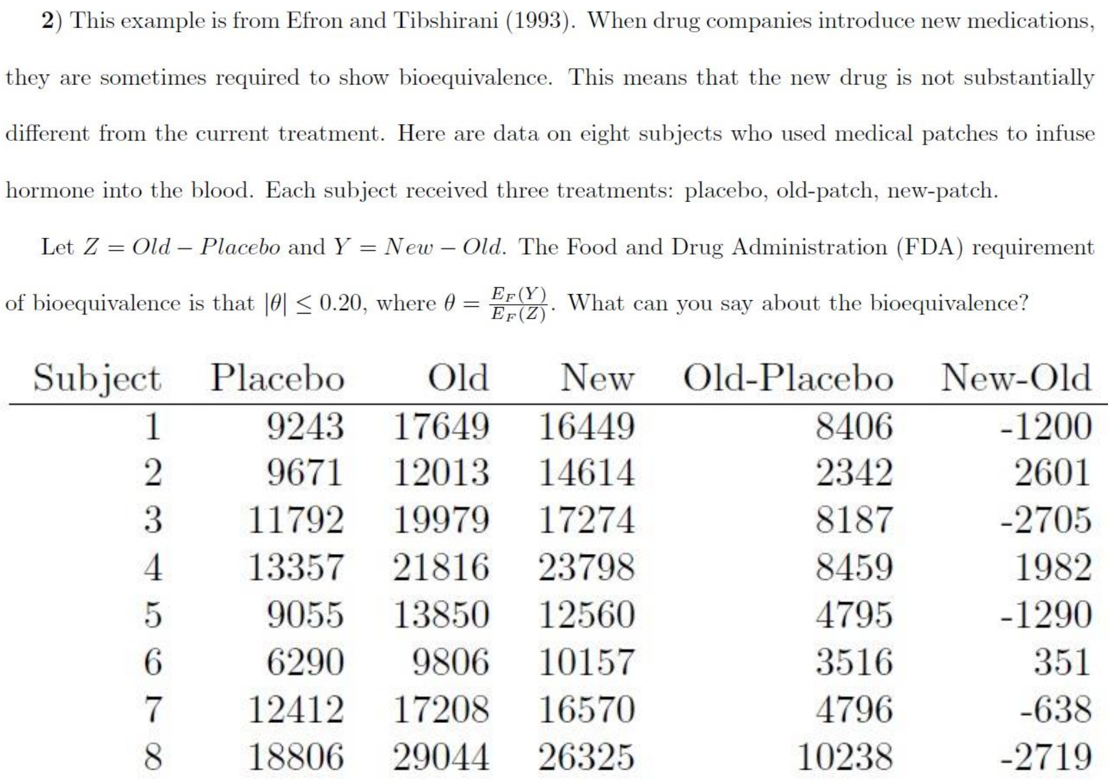
```
# (1) normal method
CI.normal.skew   = c(lower = skew.hat   + qnorm(0.025)*sd(B.skew),
                     upper = skew.hat   + qnorm(0.975)*sd(B.skew),
                     length_of_CI = qnorm(0.975)*sd(B.skew)-qnorm(0.025)*sd(B.skew))

# (2) pivotal method
CI.pivot.skew    = c(lower = 2*skew.hat   - quantile(B.skew, 0.975),
                     upper = 2*skew.hat   - quantile(B.skew, 0.025),
                     length_of_CI = - quantile(B.skew, 0.025) +  quantile(B.skew, 0.975))

# (3) percentile method
CI.percent.skew  = c(lower = quantile(B.skew, 0.025),
                     upper = quantile(B.skew, 0.975),
                     length_of_CI = quantile(B.skew, 0.975) - quantile(B.skew, 0.025))

rbind(CI.normal.skew, CI.pivot.skew, CI.percent.skew)

##                    lower    upper length_of_CI
## CI.normal.skew  1.452209 2.070288    0.6180792
## CI.pivot.skew   1.462274 2.089720    0.6274453
## CI.percent.skew 1.432778 2.060223    0.6274453
```

95% bootstrap confidence interval for median

```
# (1) normal method
CI.percent.median = c(lower = quantile(B.median, 0.025),
                      upper = quantile(B.median, 0.975),
                      length_of_CI = qnorm(0.975)*sd(B.median)-qnorm(0.025)*sd(B.median))

# (2) pivotal method
CI.pivot.median = c(lower = 2*median.hat - quantile(B.median, 0.975),
                    upper = 2*median.hat - quantile(B.median, 0.025),
                    length_of_CI = - quantile(B.median, 0.025) +  quantile(B.median,
0.975))

# (3) percentile method
CI.normal.median = c(lower = median.hat + qnorm(0.025)*sd(B.median),
                     upper = median.hat + qnorm(0.975)*sd(B.median),
                     length_of_CI =  quantile(B.median, 0.975) +  quantile(B.median,
0.025))

rbind(CI.normal.median, CI.pivot.median, CI.percent.median)
```

```
##                        lower     upper length_of_CI.97.5%
## CI.normal.median  0.1388536 0.1611464         0.30000000
## CI.pivot.median   0.1400000 0.1600000         0.02000000
## CI.percent.median 0.1400000 0.1600000         0.02229276
```

Question 02



2) This example is from Efron and Tibshirani (1993). When drug companies introduce new medications, they are sometimes required to show bioequivalence. This means that the new drug is not substantially different from the current treatment. Here are data on eight subjects who used medical patches to infuse hormone into the blood. Each subject received three treatments: placebo, old-patch, new-patch.

Let $Z = Old - Placebo$ and $Y = New - Old$. The Food and Drug Administration (FDA) requirement of bioequivalence is that $|\theta| \leq 0.20$, where $\theta = \frac{E_F(Y)}{E_F(Z)}$. What can you say about the bioequivalence?

Subject	Placebo	Old	New	Old-Placebo	New-Old
1	9243	17649	16449	8406	-1200
2	9671	12013	14614	2342	2601
3	11792	19979	17274	8187	-2705
4	13357	21816	23798	8459	1982
5	9055	13850	12560	4795	-1290
6	6290	9806	10157	3516	351
7	12412	17208	16570	4796	-638
8	18806	29044	26325	10238	-2719

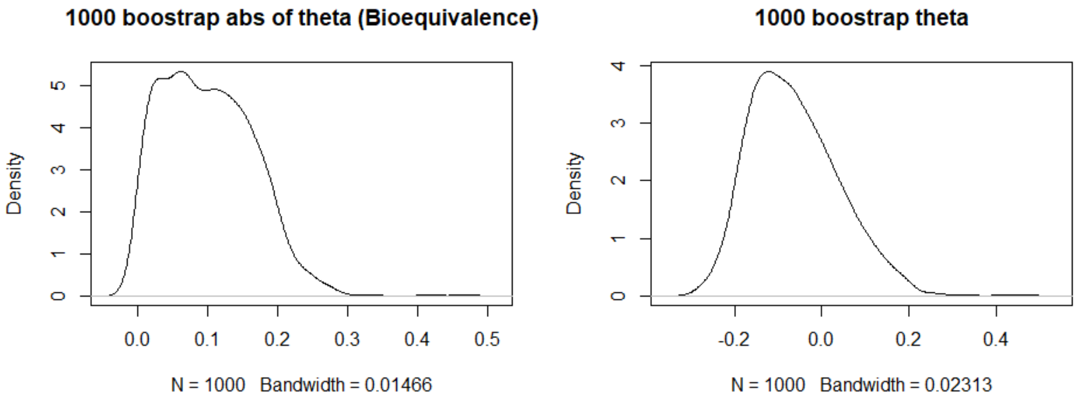
Bootstrap resampling was used 1000 times to construct the estimation of the 95% confidence interval of the bioequivalence $|\theta|$. The normal, pivotal, and percentile methods are performed to estimate the bootstrap confidence intervals.

As the definition of bioequivalence is a positive value, the percentile method is suggested to be adopted. Because this method can always produce a reasonable interval with both positive upper and lower bounds. The point estimate for bioequivalence $|\theta|$ is 0.0713061. The 95% bootstrap percentile confidence interval (0.005145321, 0.2313033) includes the FDA requirement 0.20. Hence, the bioequivalence of the new patch is not significantly substantially different from the old patch.

Note that the distribution of θ is more spread than the distribution of $|\theta|$. That is to say, the standard deviation of θ,0.1023253,is larger than the standard deviation of $|\theta|$,0.06482732. As a result, the normal and pivotal method provide a longer length of CI (0.401108/0.3776398) than the percentile method (0.226158).

```
medication = data.frame(Placebo = c( 9243,  9671, 11792, 13357,  9055,  6290, 12412, 
18806),
                            Old = c(17649, 12013, 19979, 21816, 13850,  9806, 17208, 
29044),
                            New = c(16449, 14614, 17274, 23798, 12560, 10157, 16570, 
26325))
medication$`Old-Placebo` = medication$Old - medication$Placebo
medication$`New-Old` = medication$New - medication$Old

# sample size
n = nrow(medication)
```

```
# Function of Bioequivalence
f.bio     = function(y, z) {    mean(y)/mean(z) } # theta
f.bio_abs = function(y, z) {abs(mean(y)/mean(z))} # absolute value of theta

# point estimation with absolute value of theta = Bioequivalence
bio.hat     = f.bio(medication$`New-Old`, medication$`Old-Placebo`)

# point estimation with theta
bio.hat_abs = f.bio_abs(medication$`New-Old`, medication$`Old-Placebo`)
bio.hat_abs

## [1] 0.0713061

# Bootstrap (Bioequivalence)
set.seed(20200824)
B = 1000
B.bio = B.bio_abs = numeric(length = B)
for (i in 1:B) {
  idx <- sample(1:n, size = n, replace = T)
  B.bio[i]     = f.bio(medication$`New-Old`[idx], medication$`Old-Placebo`[idx])
  B.bio_abs[i] = f.bio_abs(medication$`New-Old`[idx], medication$`Old-Placebo`[idx])
}
rm(i, idx)
```



95% bootstrap confidence interval for bioequivalence

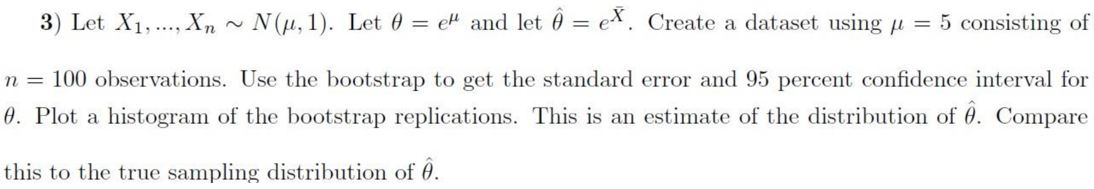
```
# Bootstrap results: absolute value of theta
summary(unlist(B.bio_abs))

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00000 0.04810 0.09601 0.10059 0.14769 0.46911

sd(unlist(B.bio_abs))

## [1] 0.06482732

# Bootstrap results: theta
summary(unlist(B.bio))

##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -0.286105 -0.139080 -0.072944 -0.062107  0.003776  0.469108

sd(unlist(B.bio))
```

```
## [1] 0.1023253
```

```
# (1) normal method
CI.normal.bio = c(lower = bio.hat_abs + qnorm(0.025)*sd(B.bio),
                  upper = bio.hat_abs + qnorm(0.975)*sd(B.bio),
                  length_of_CI = qnorm(0.975)*sd(B.bio)-qnorm(0.025)*sd(B.bio))

# (2) pivotal method
CI.pivot.bio = c(lower = 2*bio.hat_abs - quantile(B.bio, 0.975),
                 upper = 2*bio.hat_abs - quantile(B.bio, 0.025),
                 length_of_CI = - quantile(B.bio, 0.025) + quantile(B.bio, 0.975))

# (3) percentile method
CI.percent.bio = c(lower = quantile(B.bio_abs, 0.025),
                   upper = quantile(B.bio_abs, 0.975),
                   length_of_CI = quantile(B.bio_abs, 0.975)- quantile(B.bio_abs, 0.025))

rbind(CI.normal.bio, CI.pivot.bio, CI.percent.bio)
```

```
##                       lower     upper length_of_CI
## CI.normal.bio  -0.129247888 0.2718601    0.4011080
## CI.pivot.bio   -0.013192035 0.3644478    0.3776398
## CI.percent.bio  0.005145321 0.2313033    0.2261580
```

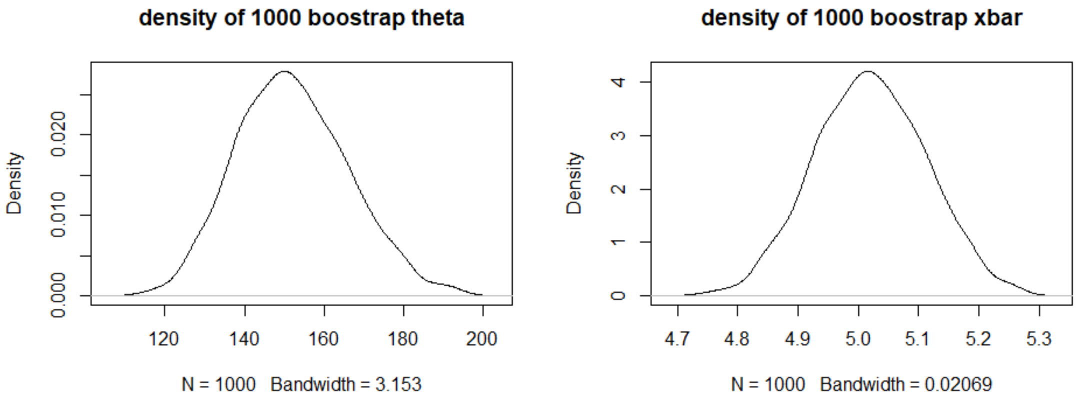
Question 03



3) Let $X_1, ..., X_n \sim N(\mu, 1)$. Let $\theta = e^{\mu}$ and let $\hat{\theta} = e^{\bar{X}}$. Create a dataset using $\mu = 5$ consisting of $n = 100$ observations. Use the bootstrap to get the standard error and 95 percent confidence interval for θ. Plot a histogram of the bootstrap replications. This is an estimate of the distribution of $\hat{\theta}$. Compare this to the true sampling distribution of $\hat{\theta}$.

Bootstrap resampling was used 1000 times to construct the estimation of the 95% confidence interval of $\hat{\theta}$. The normal, pivotal, and percentile methods are performed to estimate the bootstrap confidence intervals. The three methods produce similar results. The point estimate for $\hat{\theta}$ is 151.2321. The 95% percentile confidence interval is (127.2282, 180.2540).

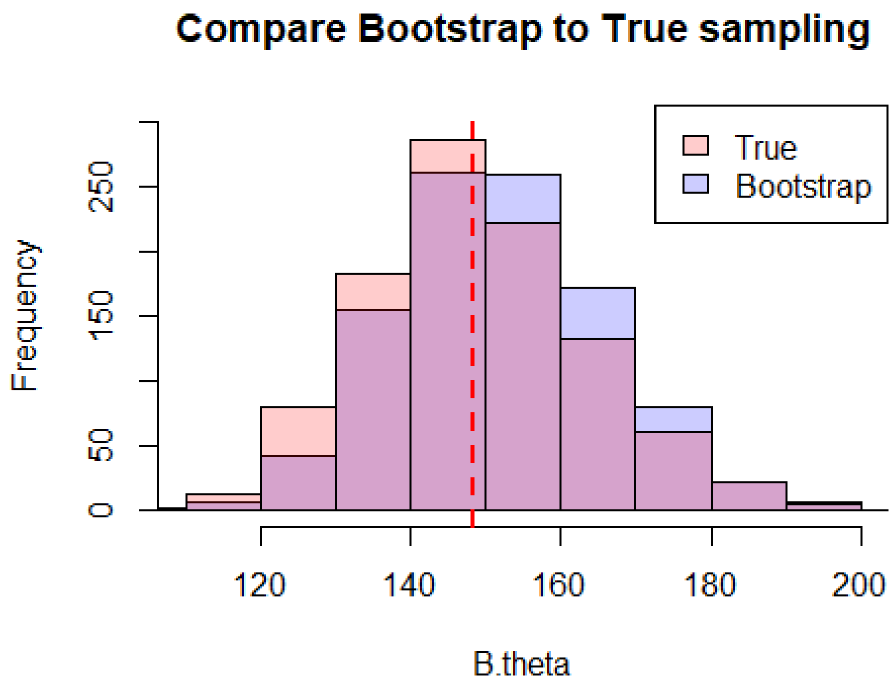
```
# Data from Normal Distribution with mean 5 and sd 1
set.seed(20201111)
x = rnorm(100, mean = 5, sd = 1)

# sample size
n = length(x)

# Function of theta
f.theta = function(x) {exp(mean(x))}
f.xbar =  function(x) {mean(x)}

# point estimation
theta.hat = f.theta(x)
theta.hat

## [1] 151.2321

# Bootstrap (theta)
B = 1000
B.theta = B.xbar = numeric(length = B)
for (i in 1:B) {
  x.new = sample(x, size = n, replace = T)
  B.theta[i] = f.theta(x.new)
  B.xbar[i]  = f.xbar(x.new)
}
rm(i, x.new)
```



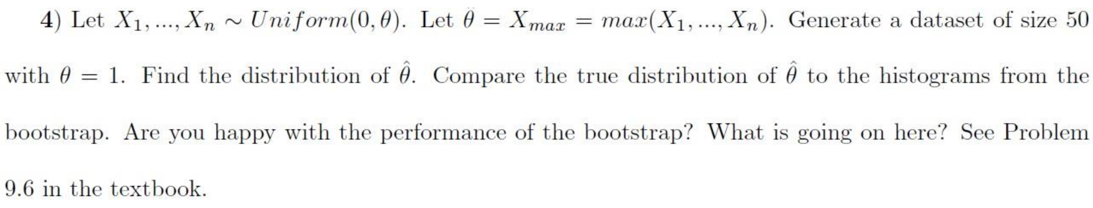
95% bootstrap confidence interval for $\hat{\theta}$

```
# (1) normal method
CI.normal.theta = c(lower = theta.hat + qnorm(0.025)*sd(B.theta),
                    upper = theta.hat + qnorm(0.975)*sd(B.theta),
                    length_of_CI = qnorm(0.975)*sd(B.theta)-qnorm(0.025)*sd(B.theta))

# (2) pivotal method
CI.pivot.theta = c(lower = 2*theta.hat - quantile(B.theta, 0.975),
                   upper = 2*theta.hat - quantile(B.theta, 0.025),
                   length_of_CI = - quantile(B.theta, 0.025) + quantile(B.theta, 0.975))

# (3) percentile method
CI.percent.theta = c(lower = quantile(B.theta, 0.025),
                     upper = quantile(B.theta, 0.975),
                     length_of_CI = quantile(B.theta, 0.975)- quantile(B.theta, 0.025))

rbind(CI.normal.theta, CI.pivot.theta, CI.percent.theta)

##                     lower    upper length_of_CI
## CI.normal.theta  123.8937 178.5705     54.67672
## CI.pivot.theta   122.2102 175.2360     53.02578
## CI.percent.theta 127.2282 180.2540     53.02578
```

The histogram below compares the bootstrap resampling distribution and the true sampling distribution. The plot below depicted the similarity between the two distributions. The majority of the true sampling distribution is covered by bootstrap resampling distribution. This indicates that the bootstrap method performs well. Note that the red dashed line indicates the true $\theta = e^5 = 148.4132$.



Question 04



4) Let $X_1, ..., X_n \sim Uniform(0, \theta)$. Let $\hat{\theta} = X_{max} = max(X_1, ..., X_n)$. Generate a dataset of size 50 with $\theta = 1$. Find the distribution of $\hat{\theta}$. Compare the true distribution of $\hat{\theta}$ to the histograms from the bootstrap. Are you happy with the performance of the bootstrap? What is going on here? See Problem 9.6 in the textbook.

It is easy to show the true sampling distribution of $\hat{\theta}$, which is the largest order statistics of Unif(0,θ),follows Beta(50,1). Here, bootstrap resampling was used 1000 times to construct the 95% confidence interval of $\hat{\theta}$. The normal, pivotal, and percentile methods are performed to estimate the bootstrap confidence intervals. The point estimate for θ is 0.9996302. The 95% percentile confidence interval is (0.9489215, 0.9996302). Obviously, the performance of the bootstrap method is poor.

```
set.seed(20200824)
theta = 1
x = runif(50, 0, theta)

# sample size
n = length(x)

# point estimation
max.hat = max(x)
max.hat

## [1] 0.9996302

# Bootstrap (max)
B = 1000
B.max = numeric(length = B)
for (i in 1:B) {
  x.new = sample(x, size = n, replace = T)
  B.max[i] = max(x.new)
}
rm(i, x.new)
```

95% bootstrap confidence interval for $\hat{\theta}$

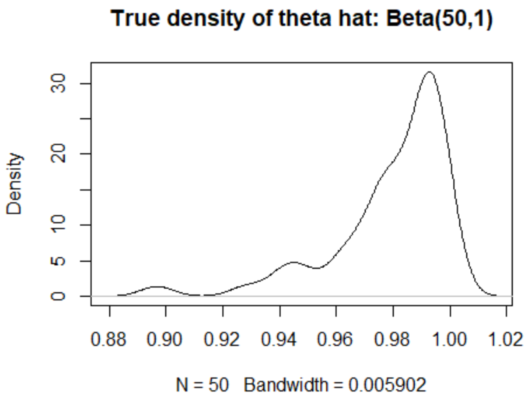
```
# (1) normal method
CI.normal.max = c(lower = max.hat + qnorm(0.025)*sd(B.max),
                  upper = max.hat + qnorm(0.975)*sd(B.max),
                  length_of_CI = qnorm(0.975)*sd(B.max)-qnorm(0.025)*sd(B.max))

# (2) pivotal method
CI.pivot.max = c(lower = 2*max.hat - quantile(B.max, 0.975),
                 upper = 2*max.hat - quantile(B.max, 0.025),
                 length_of_CI = - quantile(B.max, 0.025) + quantile(B.max, 0.975))

# (3) percentile method
CI.percent.max = c(lower = quantile(B.max, 0.025),
                   upper = quantile(B.max, 0.975),
                   length_of_CI = quantile(B.max, 0.975)- quantile(B.max, 0.025))

rbind(CI.normal.max, CI.pivot.max, CI.percent.max)
```

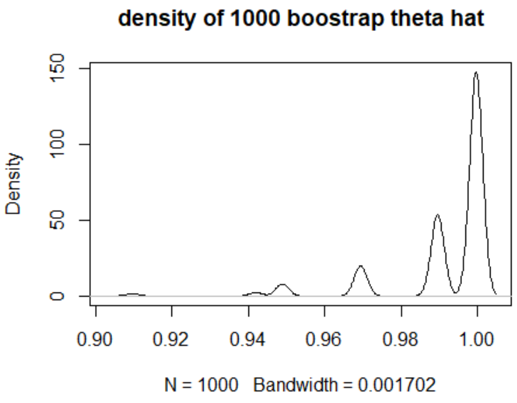
```
##                      lower     upper length_of_CI
## CI.normal.max  0.9707080 1.0285525   0.05784446
## CI.pivot.max   0.9996302 1.0503390   0.05070874
## CI.percent.max 0.9489215 0.9996302   0.05070874
```

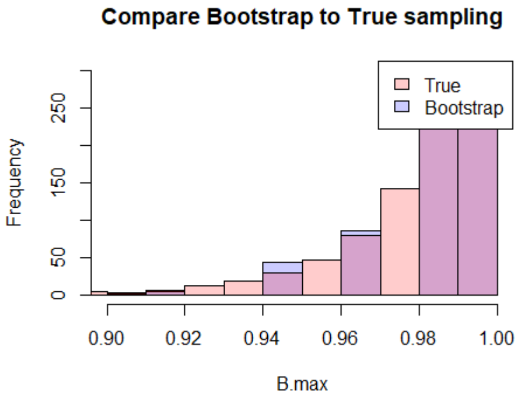
The histogram and the density plot are produced to compare the bootstrap resampling distribution and the true sampling distribution. These plots below depicted the differences between the two distributions. In the histogram, the bootstrap resampling distribution did not cover the true sampling distribution well. The bootstrap fails because it does not correctly model the true sampling distribution when the true distribution has an extremely heavy tail. Heavy-tailed distributions usually cause problems for both asymptotic and bootstrap inference. Athreya (1987) mentioned that the bootstrap method fails to converge to a deterministic distribution when the variance does not exist.

Further, the sample mean is greatly influenced by extreme observations. Hall(1990) proved that the bootstrap distribution does not converge in probability in such cases because extreme values of the sample do not converge in probability. However, those extreme values may converge in distribution, and then the bootstrap distribution converges in distribution weakly.

Arcones and Gine (1989) proposed that an alternative bootstrap's resampling strategy, m out of n bootstrap, is consistent if $m/n \to 0$ when n goes to infinity. Romano and Wolf (1999) showed that m out of n strategy also works well when the subsamples are obtained without replacement.To verify the theory, the m out of n bootstrap is performed below. The 95% percentile confidence interval is (0.8292601, 0.9996302) when (m,n) = (20,50) and the subsamples are obtained with replacement. Note that Cornea (2012) revealed that the m out of n bootstrap fails if the sample size is not very large from a simulation study.







m out of n bootstrap: 95% bootstrap confidence interval for $\hat{\theta}$

```
m_outof_n_bootstrap= function(n=50, m=10, replace = T){

set.seed(20200824)
theta = 1
x = runif(n, 0, theta)

# Bootstrap (max)
B = 1000
B.max = numeric(length = B)
for (i in 1:B) {
  x.new = sample(x, size = m, replace = replace)
  B.max[i] = max(x.new)
}

# percentile method
CI.percent.max = c(lower = quantile(B.max, 0.025),
                   upper = quantile(B.max, 0.975),
                   length_of_CI = quantile(B.max, 0.975)- quantile(B.max, 0.025))

return(CI.percent.max)
}
m_outof_n_bootstrap(n=50,m=20, replace=T)
```

```
##         lower.2.5%        upper.97.5% length_of_CI.97.5% 
##          0.8292601          0.9996302          0.1703701
```

```
m_outof_n_bootstrap(n=50,m=20, replace=F)
```

```
##         lower.2.5%        upper.97.5% length_of_CI.97.5% 
##         0.90784006         0.99963023         0.09179017
```

```
m_outof_n_bootstrap(n=500,m=20, replace=T)
```

```
##         lower.2.5%        upper.97.5% length_of_CI.97.5% 
##          0.8644505          0.9996302          0.1351798
```

```
m_outof_n_bootstrap(n=500,m=20, replace=F)
```

```
##         lower.2.5%        upper.97.5% length_of_CI.97.5% 
##          0.8728630          0.9996302          0.1267672
```

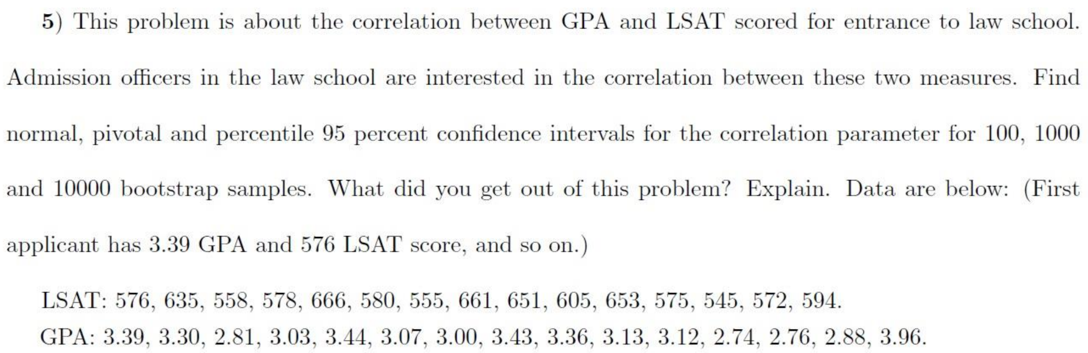
```
m_outof_n_bootstrap(n=5000,m=20, replace=T)
```

```
##         lower.2.5%        upper.97.5% length_of_CI.97.5% 
##          0.8364300          0.9986129          0.1621829
```

```
m_outof_n_bootstrap(n=5000,m=20, replace=F)
```

```
##         lower.2.5%        upper.97.5% length_of_CI.97.5% 
##          0.8357532          0.9986129          0.1628598
```

Question 05

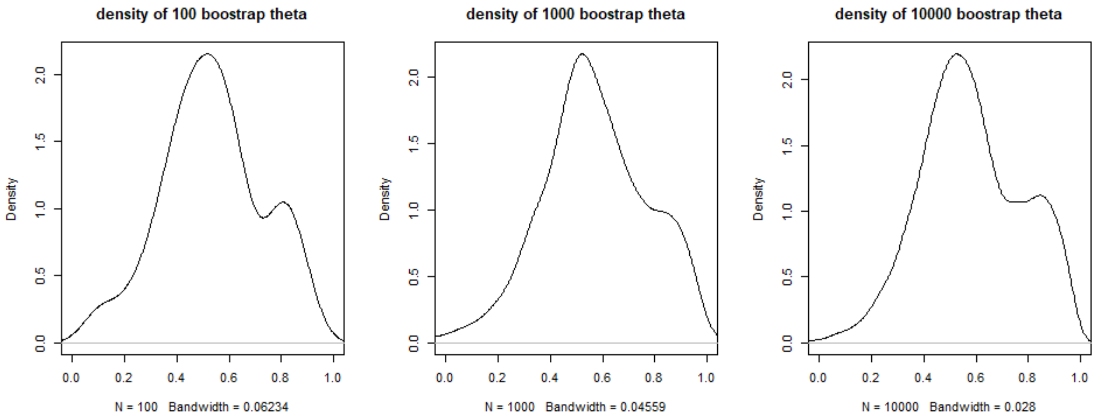


5) This problem is about the correlation between GPA and LSAT scored for entrance to law school. Admission officers in the law school are interested in the correlation between these two measures. Find normal, pivotal and percentile 95 percent confidence intervals for the correlation parameter for 100, 1000 and 10000 bootstrap samples. What did you get out of this problem? Explain. Data are below: (First applicant has 3.39 GPA and 576 LSAT score, and so on.)

LSAT: 576, 635, 558, 578, 666, 580, 555, 661, 651, 605, 653, 575, 545, 572, 594.
GPA: 3.39, 3.30, 2.81, 3.03, 3.44, 3.07, 3.00, 3.43, 3.36, 3.13, 3.12, 2.74, 2.76, 2.88, 3.96.

Bootstrap resampling was used by 100, 1000, and times to construct the 95% confidence interval of correlation between GPA and LSAT presented in the law school admission data. The normal, pivotal, and percentile methods are performed to estimate the bootstrap confidence intervals. For this dataset, the three methods produce similar results. The point estimate for the correlation is 0.5459189. The 95% percentile confidence interval from 10000 bootstrap resampling is (0.2065212,0.9382321).

```
school = data.frame( GPA = c(3.39, 3.30, 2.81, 3.03,
                             3.44, 3.07, 3.00, 3.43,
                             3.36, 3.13, 3.12, 2.74,
                             2.76, 2.88, 3.96),
                     LSAT = c(576, 635, 558, 578,
                              666, 580, 555, 661,
                              651, 605, 653, 575,
                              545, 572, 594))
# sample size
n = nrow(school)

# point estimation
cor.hat = cor(school$GPA, school$LSAT)
cor.hat

## [1] 0.5459189

# Bootstrap (correlation)
set.seed(20200824)
B = c(100, 1000, 10000)
B.cor = numeric(length = B[3])
for (i in 1:B[3]) {
  idx <- sample(1:n, size = n, replace = T)
  B.cor[i] = cor(school$GPA[idx], school$LSAT[idx])
}
rm(i, idx)
```



95% bootstrap confidence interval for the correlation ρ

```
# (1) normal method
CI.normal.cor = data.frame(B = B,
                            lower = sapply(B, function(b) cor.hat +
                                             qnorm(0.025)*sd(B.cor[1:b])),
                            upper = sapply(B, function(b) cor.hat +
                                             qnorm(0.975)*sd(B.cor[1:b])))
# (2) pivotal method
CI.pivot.cor = data.frame(B = B,
                           lower = sapply(B, function(b) 2*cor.hat -
                                            quantile(B.cor[1:b], 0.975)),
                           upper = sapply(B, function(b) 2*cor.hat -
                                            quantile(B.cor[1:b], 0.025)))

# (3) percentile method
CI.percent.cor = data.frame(B = B,
                             lower = sapply(B, function(b) quantile(B.cor[1:b], 0.025)),
                             upper = sapply(B, function(b) quantile(B.cor[1:b], 0.975)))

result = rbind(CI.normal.cor, CI.pivot.cor, CI.percent.cor)
RESULT = data.frame(method =c("normal","","","pivotal","","","percentile","",""),
                    bootstrap_times=result[,1],
                    lower=result[,2], upper = result[,3],
                    length_of_CI = result[,3] - result[,2])
RESULT
```

```
##       method bootstrap_times     lower     upper length_of_CI
## 1     normal             100 0.1699013 0.9219365    0.7520352
## 2                       1000 0.1480387 0.9437991    0.7957605
## 3                      10000 0.1611848 0.9306530    0.7694682
## 4    pivotal             100 0.2168940 0.9658398    0.7489458
## 5                       1000 0.1573236 0.9198819    0.7625583
## 6                      10000 0.1536058 0.8853167    0.7317109
## 7 percentile             100 0.1259981 0.8749438    0.7489458
## 8                       1000 0.1719559 0.9345143    0.7625583
## 9                      10000 0.2065212 0.9382321    0.7317109
```

The small sample size, 15, leads to the broader confidence interval, around 0.7317-0.7958. If the sample size increases, a narrower confidence interval is produced. The simulation study below for different sample size 100,1000, and 10000 verifies the relationship between sample size and confidence interval.

```r
SampleSize = function(size=1000){
GPA  = rnorm(n = size, mean = mean(school$GPA), sd = sd(school$GPA))
LSAT = rnorm(n = size, mean = mean(school$LSAT), sd = sd(school$LSAT))
school = data.frame(GPA=GPA, LSAT=LSAT)

# sample size
n = nrow(school)

# point estimation
cor.hat = cor(school$GPA, school$LSAT)
cor.hat

# Bootstrap (correlation)
set.seed(20200824)
B = c(100, 1000, 10000)
B.cor = numeric(length = B[3])
for (i in 1:B[3]) {
  idx <- sample(1:n, size = n, replace = T)
  B.cor[i] = cor(school$GPA[idx], school$LSAT[idx])
}
rm(i, idx)

# (1) normal method
CI.normal.cor = data.frame(B = B,
                            lower = sapply(B, function(b) cor.hat +
                                             qnorm(0.025)*sd(B.cor[1:b])),
                            upper = sapply(B, function(b) cor.hat +
                                             qnorm(0.975)*sd(B.cor[1:b])))
# (2) pivotal method
CI.pivot.cor = data.frame(B = B,
                           lower = sapply(B, function(b) 2*cor.hat -
                                            quantile(B.cor[1:b], 0.975)),
                           upper = sapply(B, function(b) 2*cor.hat -
                                            quantile(B.cor[1:b], 0.025)))
# (3) percentile method
CI.percent.cor = data.frame(B = B,
                             lower = sapply(B, function(b) quantile(B.cor[1:b], 0.025)),
                             upper = sapply(B, function(b) quantile(B.cor[1:b], 0.975)))

result = rbind(CI.normal.cor, CI.pivot.cor, CI.percent.cor)
RESULT = data.frame(method =c("normal","","","pivotal","","","percentile","",""),
                    bootstrap_times=result[,1],
                    lower=result[,2], upper = result[,3],
                    length_of_CI = result[,3] - result[,2])
return(RESULT)
}

SampleSize(size=100)
```

```
##       method bootstrap_times      lower      upper length_of_CI
## 1     normal             100 -0.1816826 0.10246729    0.2841499
## 2                       1000 -0.2204749 0.14125951    0.3617344
## 3                      10000 -0.2285635 0.14934815    0.3779117
## 4    pivotal            100 -0.1771958 0.07859247    0.2557883
## 5                       1000 -0.2208384 0.13576903    0.3566074
## 6                      10000 -0.2293067 0.14650616    0.3758128
## 7 percentile            100 -0.1578078 0.09798046    0.2557883
## 8                       1000 -0.2149844 0.14162303    0.3566074
## 9                      10000 -0.2257215 0.15009130    0.3758128
```

```r
SampleSize(size=1000)
```

```
##       method bootstrap_times      lower       upper length_of_CI
## 1     normal             100 -0.1054550 0.005128252    0.1105833
## 2                       1000 -0.1074331 0.007106360    0.1145395
## 3                      10000 -0.1078175 0.007490779    0.1153083
## 4    pivotal             100 -0.1042775 0.004696268    0.1089738
## 5                       1000 -0.1052532 0.006223823    0.1114770
## 6                      10000 -0.1079067 0.006377761    0.1142844
## 7 percentile             100 -0.1050230 0.003950759    0.1089738
## 8                       1000 -0.1065506 0.004926415    0.1114770
## 9                      10000 -0.1067045 0.007579907    0.1142844
```

```r
SampleSize(size=10000)
```

```
##       method bootstrap_times       lower      upper length_of_CI
## 1     normal             100 -0.02474387 0.01609144   0.04083531
## 2                       1000 -0.02363854 0.01498611   0.03862466
## 3                      10000 -0.02403835 0.01538592   0.03942426
## 4    pivotal             100 -0.02386299 0.01813371   0.04199669
## 5                       1000 -0.02395020 0.01431474   0.03826494
## 6                      10000 -0.02368134 0.01563676   0.03931810
## 7 percentile             100 -0.02678614 0.01521056   0.04199669
## 8                       1000 -0.02296716 0.01529777   0.03826494
## 9                      10000 -0.02428919 0.01502891   0.03931810
```