HW3

Tori Wang

2023-04-27

```
source("bootsample.R")
source("bootstats.R")
library(MASS)
data("cars")
summary(cars)
```

```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00  
```

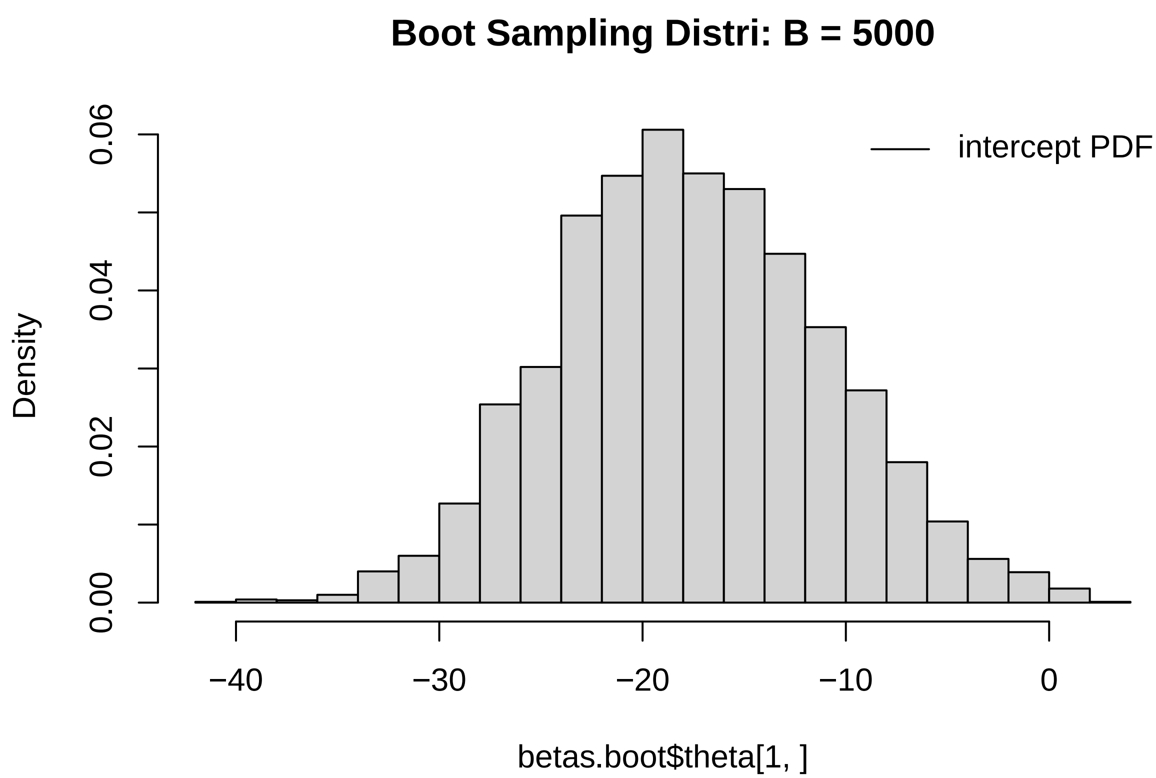
Problem 1

Part a)

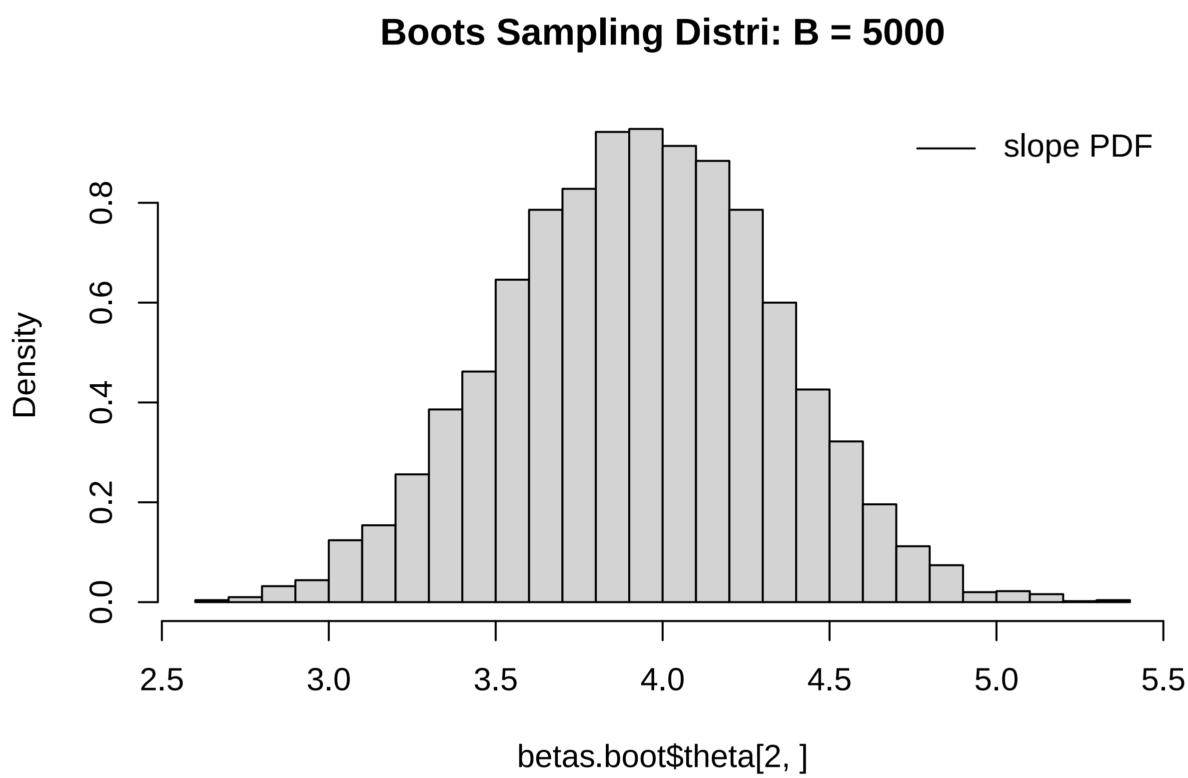
Use the parametric bootstrap to construct the following bootstrap CIs for the intercept B1 and the slope B1, at level a = 0.05.

```
set.seed(1)
x = cars$speed
y = cars$dist
B=5000

alpha = 0.05

#calculated least squares estimate betas
betas = lm(y~x)
betas.summary=summary(betas)

# store the intrcept and slope for use in CIs
beta.int=betas.summary$coefficients[1,1]
beta.slope=betas.summary$coefficients[2,1]

#obtain predicted values
yhat = betas$fitted.values

#obtain residuals
boot.samples = bootsampling(betas$residuals,B)
y.bootsamples = matrix(yhat,length(yhat),ncol(boot.samples)) + boot.samples
```

```r
betahat = function(y,x) {lm(y~x)$coefficients}

# calculate the bootstrap beta coefficients
# using our function boot.stats

#bootsampling the coefficients
betas.boot = boot.stats(y.bootsamples,betahat,x=x)

hist(betas.boot$theta[1,], breaks=20,freq=F, main=paste("Boot Sampling Distri: B =",B))
  legend("topright",expression("intercept PDF"),lty=1,bty="n")
```




```r
hist(betas.boot$theta[2,], breaks=20,freq=F, main=paste("Boots Sampling Distri: B =",B))
  legend("topright",expression("slope PDF"),lty=1,bty="n")
```



1. Normal Bootstrap CI

```
parametric_int_CI <- matrix(nrow = 4, ncol = 2)
parametric_slope_CI <- matrix(nrow = 4, ncol = 2)
zval = qnorm(p=alpha/2, lower.tail=FALSE)

normal.CI.int= cbind(beta.int-zval*sqrt(betas.boot$cov[1,1]),
                            beta.int+zval*sqrt(betas.boot$cov[1,1]))
normal.CI.slope=cbind(beta.slope-zval*sqrt(betas.boot$cov[2,2]),
                            beta.slope+zval*sqrt(betas.boot$cov[2,2]))

parametric_int_CI[1,] <- normal.CI.int
parametric_slope_CI[1,] <- normal.CI.slope
```

2. Basic Bootstrap CI

```
basic.CI.int=cbind(2*beta.int-quantile(betas.boot$theta[1,],probs=(1-alpha/2)),
                   2*beta.int-quantile(betas.boot$theta[1,],probs=(alpha/2)))
basic.CI.slope=cbind(2*beta.slope-quantile(betas.boot$theta[2,],probs=(1-alpha/2)),
                   2*beta.slope-quantile(betas.boot$theta[2,],probs=(alpha/2)))

parametric_int_CI[2,] <- basic.CI.int
parametric_slope_CI[2,] <- basic.CI.slope
```

3. **Percentile Bootstrap CI**

```
percentile.CI.int=cbind(quantile(betas.boot$theta[1,],probs=(alpha/2)),
                  quantile(betas.boot$theta[1,],probs=(1-alpha/2)))
percentile.CI.slope=cbind(quantile(betas.boot$theta[2,],probs=(alpha/2)),
                  quantile(betas.boot$theta[2,],probs=(1-alpha/2)))

parametric_int_CI[3,] <- percentile.CI.int
parametric_slope_CI[3,] <- percentile.CI.slope
```

4. **Bias corrected Bootstrap CI**

```
z0int = qnorm(sum(betas.boot$theta[1,]<beta.int)/B)
z0slope = qnorm(sum(betas.boot$theta[2,]<beta.slope)/B)
A1int= pnorm(2*z0int+qnorm(alpha/2))
A2int= pnorm(2*z0int+qnorm(1-alpha/2))
A1slope= pnorm(2*z0slope+qnorm(alpha/2))
A2slope= pnorm(2*z0slope+qnorm(1-alpha/2))

BC.CI.int=cbind(quantile(betas.boot$theta[1,],probs=(A1int)),
                        quantile(betas.boot$theta[1,],probs=(A2int)) )

BC.CI.slope=cbind(quantile(betas.boot$theta[2,],probs=(A1slope)),
                                quantile(betas.boot$theta[2,],probs=(A2slope)) )

parametric_int_CI[4,] <- BC.CI.int
parametric_slope_CI[4,] <- BC.CI.slope
```

Calculate the length and the shape of each type of Bootstrap CI and report them as well.

The Parametric Bootstrap Confidence for the **Intercept** for Normal, Basic, Percentile, and Bias Corrected Bootstrap are:

```
parametric_int_CI
```

```
##           [,1]      [,2]
## [1,] -30.50036 -4.657827
## [2,] -30.80885 -5.337809
## [3,] -29.82038 -4.349344
## [4,] -29.31942 -3.723964
```

The corresponding lengths are:

```
abs(parametric_int_CI[,1] - parametric_int_CI[,2])
```

```
## [1] 25.84253 25.47104 25.47104 25.59546
```

The Parametric Bootstrap Confidence for the **Slope** for Normal, Basic, Percentile, and Bias Corrected Bootstrap are:

```
parametric_slope_CI
```

```
##          [,1]     [,2]
## [1,] 3.141782 4.723035
## [2,] 3.168833 4.738637
## [3,] 3.126181 4.695985
## [4,] 3.123355 4.689342
```

The corresponding lengths are:

```
abs(parametric_slope_CI[,1] - parametric_slope_CI[,2])
```

```
## [1] 1.581253 1.569804 1.569804 1.565987
```

Discussion

I decided to use B=5000 replicates. This is because I tested using B=1000 beforehand, however the resulting distribution was not close enough to a smooth normal distribution. As seen by the bootstrapped sampling distribution shown below **part a**, the sampling distributions using B=5000 replicates follow a noraml distribution.

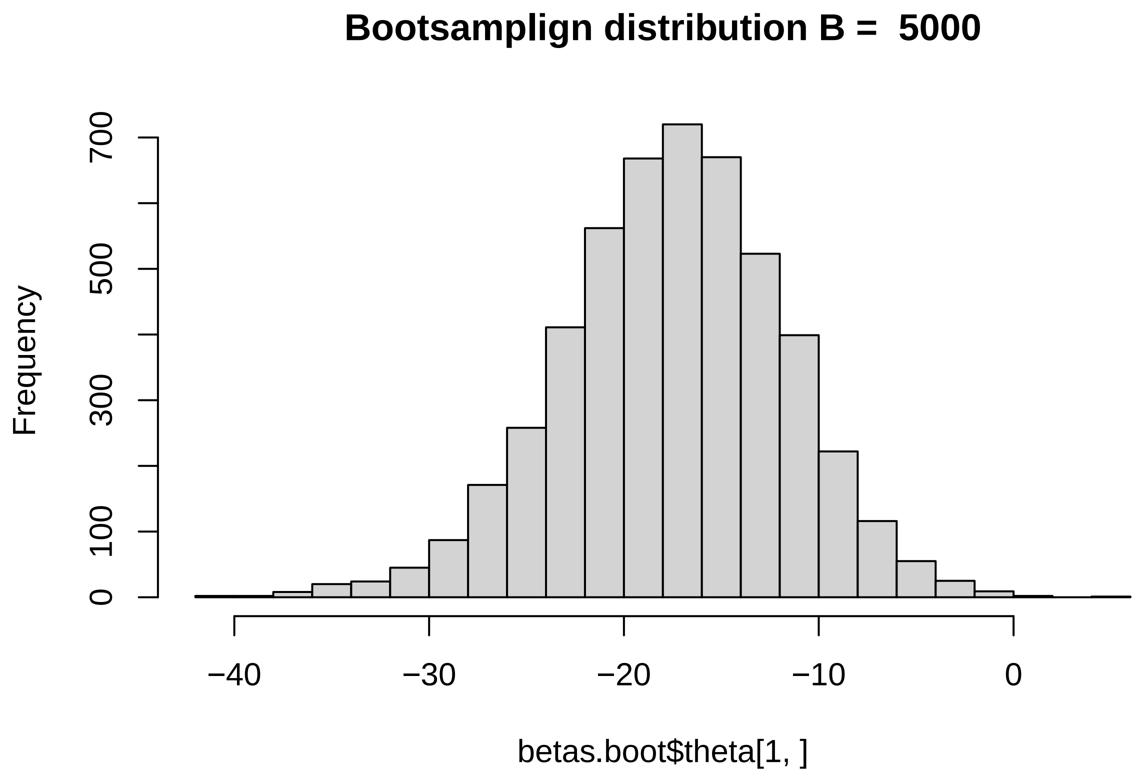
The resulting Confidence interval Lengths for the **Intercept** for Normal, Basic, Percentile, and Bias Corrected Bootstrap are: 25.84253, 25.47104, 25.47104, 25.59546. All of the confidence intervals are centered around -17.57909, and the lengths for the the Basic and Percentiles are the shortest at 25.47104.

The resulting Confidence interval Lengths for the **Slope** for Normal, Basic, Percentile, and Bias Corrected Bootstrap are: 1.581253, 1.569804, 1.569804, 1.565987. The intervals were centered around 3.906349 and the shortest length interval was the Bias Corrected Bootstrap Interval. The longest interval came from the Normal confidence interval.

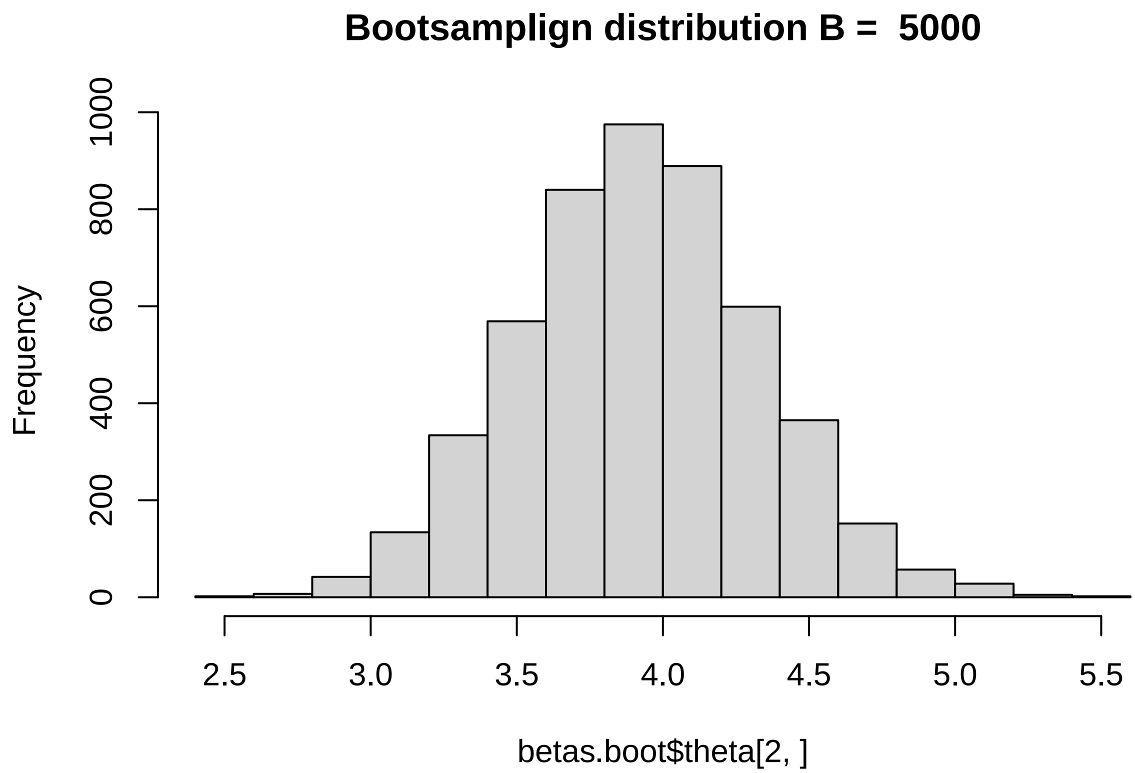
Part (b):

Use the nonparametric bootstrap to construct the following bootstrap CIs the intercept B0 and the slope B1, at level a = 0.05.

```
B= 5000
z.data = cbind(y,x)

set.seed(1)
boot.samples = bootsampling ( z.data , B)
# create function to obtain the beta coefficients
betahat = function (z) {lm (z[,1]~ z [,2])$coefficients }
# calculate the bootstrap beta coefficients
# using our function boot . stats
betas.boot = boot.stats(boot.samples, betahat)

hist(betas.boot$theta[1,], breaks = 20, main = paste("Bootsamplign distribution B = ", B))
```




```
hist(betas.boot$theta[2,], breaks = 20, main = paste("Bootsamplign distribution B = ", B))
```



1. Normal Bootstrap CI

```
alpha = 0.05
nonparametric_int_CI <- matrix(nrow = 4, ncol = 2)
nonparametric_slope_CI <- matrix(nrow = 4, ncol = 2)

zval = qnorm(p=alpha/2, lower.tail=FALSE)
normal.CI.int= cbind(beta.int-zval*sqrt(betas.boot$cov[1,1]),
                            beta.int+zval*sqrt(betas.boot$cov[1,1]))
normal.CI.slope=cbind(beta.slope-zval*sqrt(betas.boot$cov[2,2]),
                            beta.slope+zval*sqrt(betas.boot$cov[2,2]))

nonparametric_int_CI[1,] <- normal.CI.int
nonparametric_slope_CI[1,] <- normal.CI.slope
```

2. Basic Bootstrap CI

```
basic.CI.int=cbind(2*beta.int-quantile(betas.boot$theta[1,],probs=(1-alpha/2)),
                   2*beta.int-quantile(betas.boot$theta[1,],probs=(alpha/2)))
basic.CI.slope=cbind(2*beta.slope-quantile(betas.boot$theta[2,],probs=(1-alpha/2)),
                   2*beta.slope-quantile(betas.boot$theta[2,],probs=(alpha/2)))

nonparametric_int_CI[2,] <- basic.CI.int
nonparametric_slope_CI[2,] <- basic.CI.slope
```

3. Percentile Bootstrap CI

```
percentile.CI.int=cbind(quantile(betas.boot$theta[1,],probs=(alpha/2)),
                  quantile(betas.boot$theta[1,],probs=(1-alpha/2)))
percentile.CI.slope=cbind(quantile(betas.boot$theta[2,],probs=(alpha/2)),
                  quantile(betas.boot$theta[2,],probs=(1-alpha/2)))

nonparametric_int_CI[3,] <- percentile.CI.int
nonparametric_slope_CI[3,] <- percentile.CI.slope
```

4. Bias corrected Bootstrap CI

```
z0int = qnorm(sum(betas.boot$theta[1,]<beta.int)/B)
z0slope = qnorm(sum(betas.boot$theta[2,]<beta.slope)/B)
A1int= pnorm(2*z0int+qnorm(alpha/2))
A2int= pnorm(2*z0int+qnorm(1-alpha/2))
A1slope= pnorm(2*z0slope+qnorm(alpha/2))
A2slope= pnorm(2*z0slope+qnorm(1-alpha/2))

BC.CI.int=cbind(quantile(betas.boot$theta[1,],probs=(A1int)),
                        quantile(betas.boot$theta[1,],probs=(A2int)) )

BC.CI.slope=cbind(quantile(betas.boot$theta[2,],probs=(A1slope)),
                                quantile(betas.boot$theta[2,],probs=(A2slope)) )

nonparametric_int_CI[4,] <- BC.CI.int
nonparametric_slope_CI[4,] <- BC.CI.slope
```

The Nonparametric Bootstrap Confidence for the **Intercept** for Normal, Basic, Percentile, and Bias Corrected Bootstrap are:

```
nonparametric_int_CI
```

```
##           [,1]      [,2]
## [1,] -28.83567 -6.322521
## [2,] -28.30158 -5.966361
## [3,] -29.19183 -6.856606
## [4,] -29.96588 -7.257882
```

and their corresponding lengths are:

```
abs(nonparametric_int_CI[,1] - nonparametric_int_CI[,2])
```

```
## [1] 22.51315 22.33522 22.33522 22.70800
```

The Nonparametric Bootstrap Confidence for the **Slope** for Normal, Basic, Percentile, and Bias Corrected Bootstrap are:

```
nonparametric_slope_CI
```

```
##          [,1]     [,2]
## [1,] 3.129013 4.735804
## [2,] 3.130796 4.738868
## [3,] 3.125949 4.734021
## [4,] 3.153615 4.766391
```

and their corresponding lengths are:

```r
abs(nonparametric_slope_CI[,1] - nonparametric_slope_CI[,2])
```

```
## [1] 1.606791 1.608072 1.608072 1.612775
```

Discussion

I decided to use B=5000 replicates. This is because I tested using B=1000 beforehand, however the resulting distribution was not close enough to a smooth normal distribution. As seen by the bootstrapped sampling distribution shown below **part b**, the sampling distributions using B=5000 replicates follow a normal distribution.

The lengths for the **Intercept** Confidence Intervals for Normal, Basic, Percentile, and Bias Corrected Bootstrap are: 22.51315, 22.33522, 22.33522, 22.70800. The intervals are all centered around -17.57909 and the shortest interval are the Basic and Percentile Intervals and the longest is the Bias Corrected Bootstrap.

The lengths for the **Slope** Confidence Intervals for Normal, Basic, Percentile, and Bias Corrected Bootstrap are: 1.606791, 1.608072, 1.608072, 1.612775. The Intervals are centered around 3.960003 and the longest interval is the Normal Confidence Interval with the longest being the Bias Corrected Bootstrap.

Part c)

Compare the lengths and shapes for each type of bootstrap CI you constructed in Parts (a) and (b) and comments on the results.

Each type of confidence interval for parametric and nonparametric bootstraps for the Intercept are shown below:

```r
parametric_int_CI
```

```
##           [,1]      [,2]
## [1,] -30.50036 -4.657827
## [2,] -30.80885 -5.337809
## [3,] -29.82038 -4.349344
## [4,] -29.31942 -3.723964
```

```r
nonparametric_int_CI
```

```
##           [,1]      [,2]
## [1,] -28.83567 -6.322521
## [2,] -28.30158 -5.966361
## [3,] -29.19183 -6.856606
## [4,] -29.96588 -7.257882
```

```r
abs(parametric_int_CI[,1] - parametric_int_CI[,2]) - abs(nonparametric_int_CI[,1] - nonparametric_int_C
```

```
## [1] 3.329387 3.135815 3.135815 2.887463
```

By comparing the nonparametric and parametric confidence intervals shown above for the **Intercept**, we can see that in general, the nonparametric bootstraps have shorter length than the parametric bootstrap. When we look at the differences in length, the parametric bootstrap for Normal, Basic, Percentile, and Bias Corrected Interval are longer than the nonparametric by: 3.329387, 3.135815, 3.135815, and 2.887463.

Each type of confidence interval for parametric and nonparametric bootstraps for the Slope are shown below:

```r
parametric_slope_CI
```

```
##          [,1]     [,2]
## [1,] 3.141782 4.723035
## [2,] 3.168833 4.738637
```

```
## [3,] 3.126181 4.695985
## [4,] 3.123355 4.689342
```

```r
nonparametric_slope_CI
```

```
##          [,1]     [,2]
## [1,] 3.129013 4.735804
## [2,] 3.130796 4.738868
## [3,] 3.125949 4.734021
## [4,] 3.153615 4.766391
```

```r
abs(parametric_slope_CI[,1] - parametric_slope_CI[,2]) - abs(nonparametric_slope_CI[,1] - nonparametric_
```

```
## [1] -0.02553855 -0.03826756 -0.03826756 -0.04678893
```

By comparing the nonparametric and parametric confidence intervals shown above for the **Slope**, we can see that in general, the parametric bootstraps have shorter length than the parametric bootstrap. When we look at the differences in length, the nonparametric bootstrap for Normal, Basic, Percentile, and Bias Corrected Interval are longer than the parametric by: 0.02553855, 0.03826756, 0.03826756, and 0.04678893.

These results are interesting because the parametric bootstraps for Intercept are longer but the nonparametric bootstraps for slopes are longer. We also observe that the differences in lengths for the intercept intervals are much larger than the differences in lengths for the slope intervals. Part of this can be attributed to the greater value of the center for the slope and intecept intervals, however it is still interesting that the slope intervals for parametric and nonparametric bootstraps have reverse patterns.