Bootstrapping Procedures ‘by hand’

A comparison of small and large data sets when bootstrapping.

Michael Ingram

November 13, 2016

Abstract

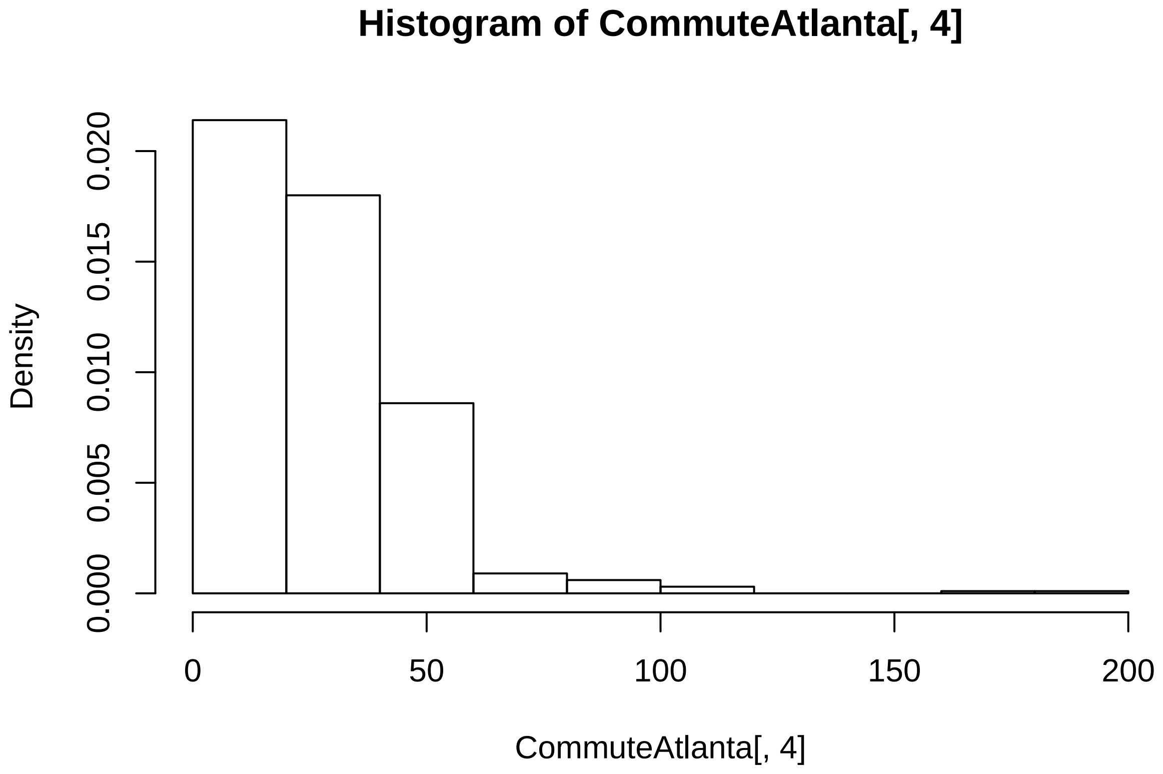
This code is a comparison using the Atlanta Commute Time data set in R. The first procedure is done with the entire data and the second with a small sample taken from the Atlanta Commute Time data set. Although there are already bootstrapping packages in R, this code is done ‘by hand’ or without the provided library. In this code, I calculate the sampling distribution of the mean, bootstrap to estimate the standard error of the mean, bootstrap to estimate the bias of the mean and then calculate 4 different types of 95% Confidence Intervals. One Interval is calculated without bootstrapping and the other three are calculated using bootstrapping. All of these procedures are then repeated again for a small sample taken from the Atlanta Commute time data set. I then compare the differences between the large and small data set and the different types of confidence intervalas used.

First draw a histogram to see what the data looks like. Notice it is right skewed.

```
library(Lock5Data)
data(CommuteAtlanta)
library(boot)
```

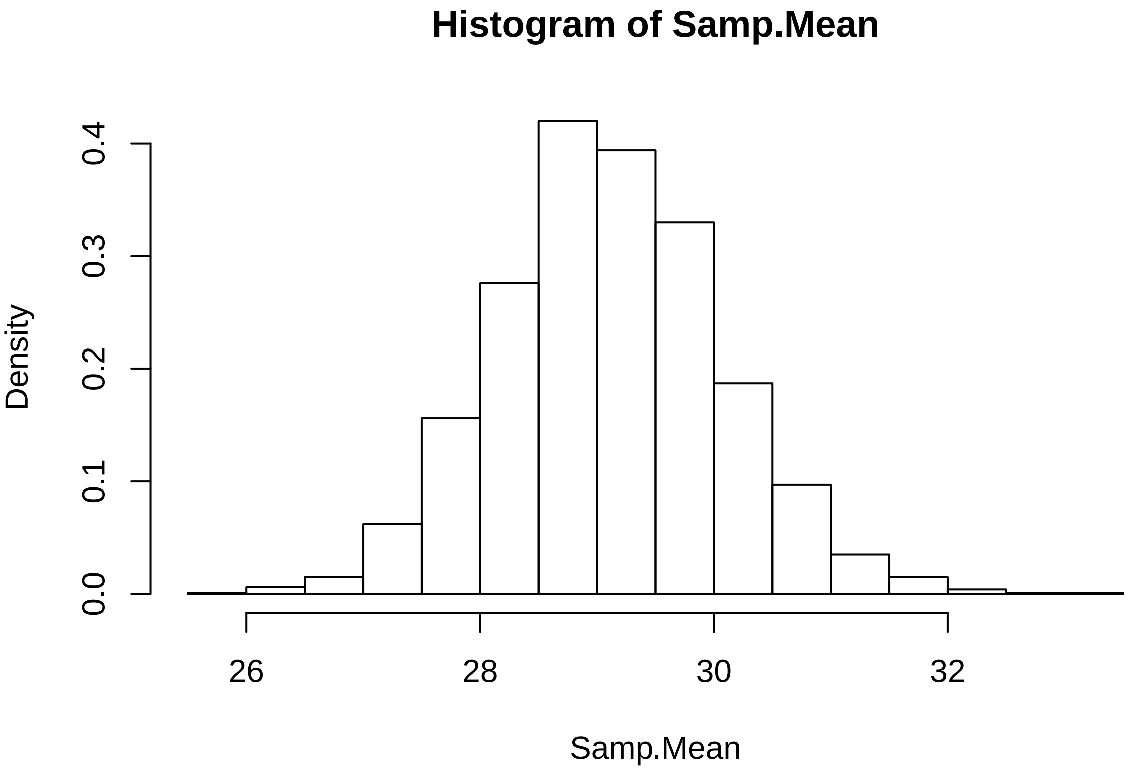
Note: I have used set.seed(1) throughout this code for comparison.

```
set.seed(1)
hist(CommuteAtlanta[,4],prob=TRUE)
```

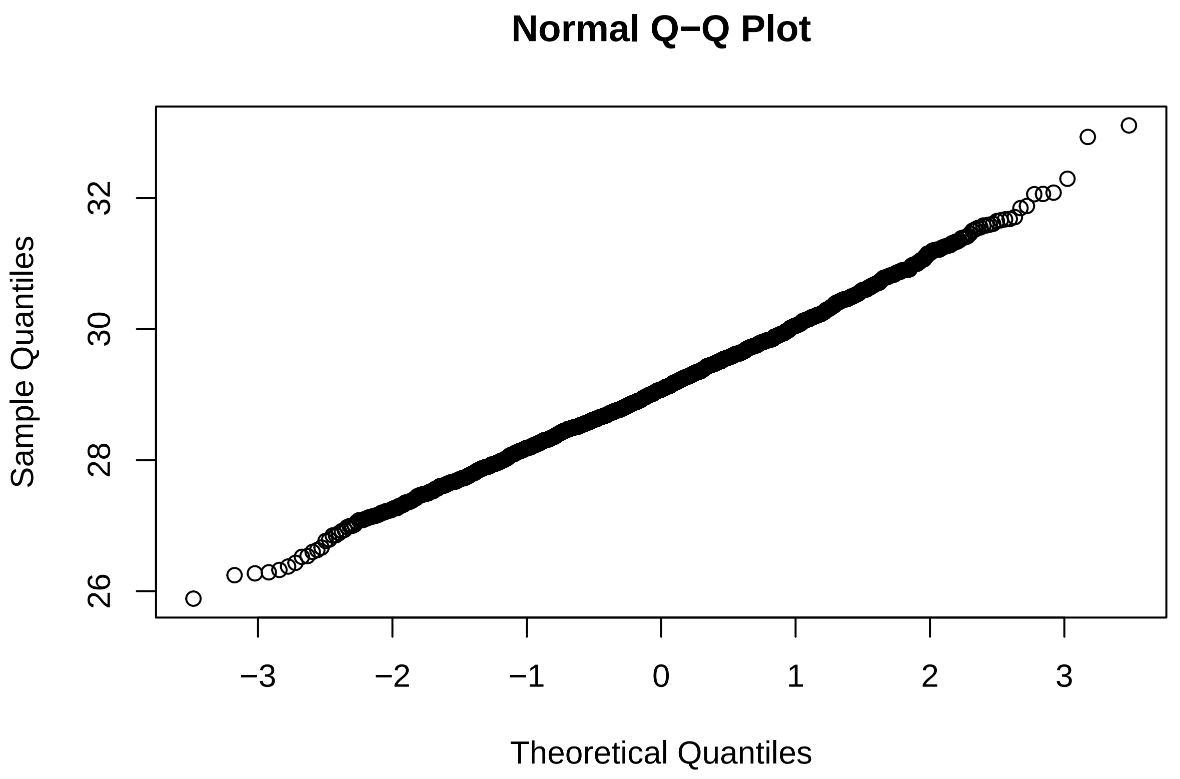


Now I'm going to approximate the sampling distribution of the mean with 2000 bootstrap samples. Then I'm going to make a qq plot to check for normality.

```
set.seed(1)
bootsamps<-matrix(0,nrow=2000, ncol=length(CommuteAtlanta[,4]))
for(i in 1:2000){
  bootsamps[i,]=sample(CommuteAtlanta[,4], replace=TRUE)

}
Samp.Mean<-apply(bootsamps,1,mean)

hist(Samp.Mean, prob=TRUE)
```



```
qqnorm(Samp.Mean)
```



Notice, the histogram and qqplot shows that the bootstrap distribution is approximately normal

I can also use bootstrapping to estimate the standard error of the mean of commute times

```
set.seed(1)
se.comutetime<-sd(Samp.Mean)
se.comutetime
```

```
## [1] 0.9603991
```

Similarly, we can use bootstrapping to estimate the bias of the mean of commute times.

```
set.seed(1)
theta.hat<-mean(CommuteAtlanta[,4])
bias.comutetime<-mean(Samp.Mean-theta.hat)
bias.comutetime
```

```
## [1] 5e-04
```

Here I compute 4 different 95% Confidence Intervals for Student's T (no bootstrapping), Standard Normal Bootstrap CI, Percentile Bootstrap CI, Bootstrap T and compare.

```
set.seed(1)
## Students T CI (no bootstrap)
xbar<-mean(CommuteAtlanta[,4])
tcrit<-qt(.975, df=499)
se.Comm<-sd(CommuteAtlanta[,4])/sqrt(500)
```

```
StudTCI<-xbar+c(-1,1)*tcrit*se.Comm

## Standard Normal CI
set.seed(1)
StdNormCI<-xbar+c(-1,1)*qnorm(.975)*se.comutetime
StdNormCI
```

```
## [1] 27.22765 30.99235
```

```
## Percentile
set.seed(1)
PercentCI<-quantile(Samp.Mean, c(.025, .975))
PercentCI
```

```
##    2.5%   97.5%
## 27.2953 31.0671
```

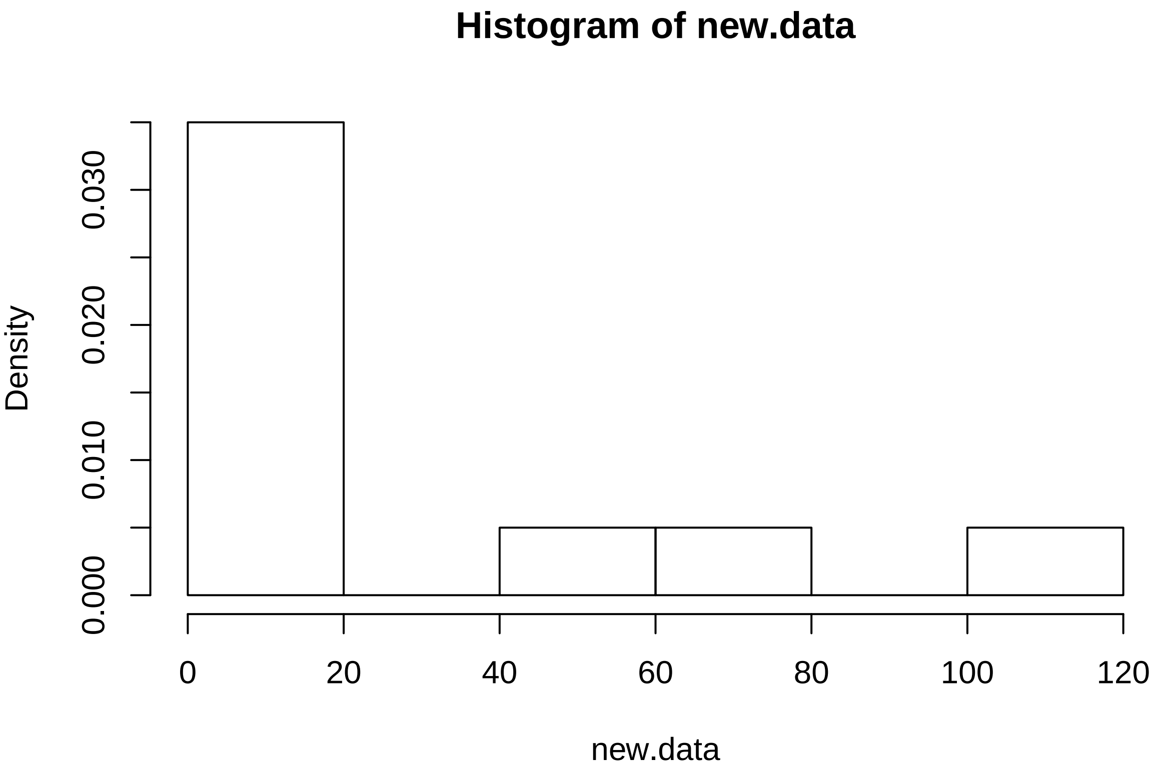
```
## Bootstrap T
set.seed(1)
boot.t.ci<-function(x, B=2000, R=500, level=.95, statistic){
  x<-as.matrix(x)
  n<-nrow(x)
  stat<-numeric(B)
  se<-numeric(B)

  boot.se<-function(x,R,f){
    x<-as.matrix(x)
    m<-nrow(x)
    th<-replicate(R, expr = {i<-sample(1:m, size=m, replace=TRUE)
    f(x[i,])
    })
    return(sd(th))
  }

  for(b in 1:B){
    j<-sample(1:n, size=n, replace=TRUE)
    y<-x[j,]
    stat[b]<-statistic(y)
    se[b]<-boot.se(y, R=R, f=statistic)
  }
  stat0<-statistic(x)
  t.stats<-(stat-stat0)/se
  se0<-sd(stat)
  alpha<- 1-level
  Qt<-quantile(t.stats,c(alpha/2, 1-(alpha/2)),type=1)
  names(Qt)<-rev(names(Qt))
  CI<-rev(stat0-Qt*se0)

}
stat<-my.mean
BootTCI<-boot.t.ci(CommuteAtlanta$Time, statistic = stat, B=2000, R=100)

## Table of Results
Table.Colname<-c("Lower Bound", "Upper Bound")
rbind(Table.Colname,StudTCI,StdNormCI, PercentCI,BootTCI)
```

```
##                2.5%               97.5%
## Table.Colname "Lower Bound"      "Upper Bound"
## StudTCI       "27.2895779777766" "30.9304220222234"
## StdNormCI     "27.2276524124941" "30.9923475875059"
## PercentCI     "27.2953"          "31.0671"
## BootTCI       "27.3267072829859" "31.171059184284"
```

Now I'm going to repeat the same procedures using a small sample of the data I used above.

```
new.data<-c(1,3 ,5,7,10,12,15, 45, 75, 120)
```

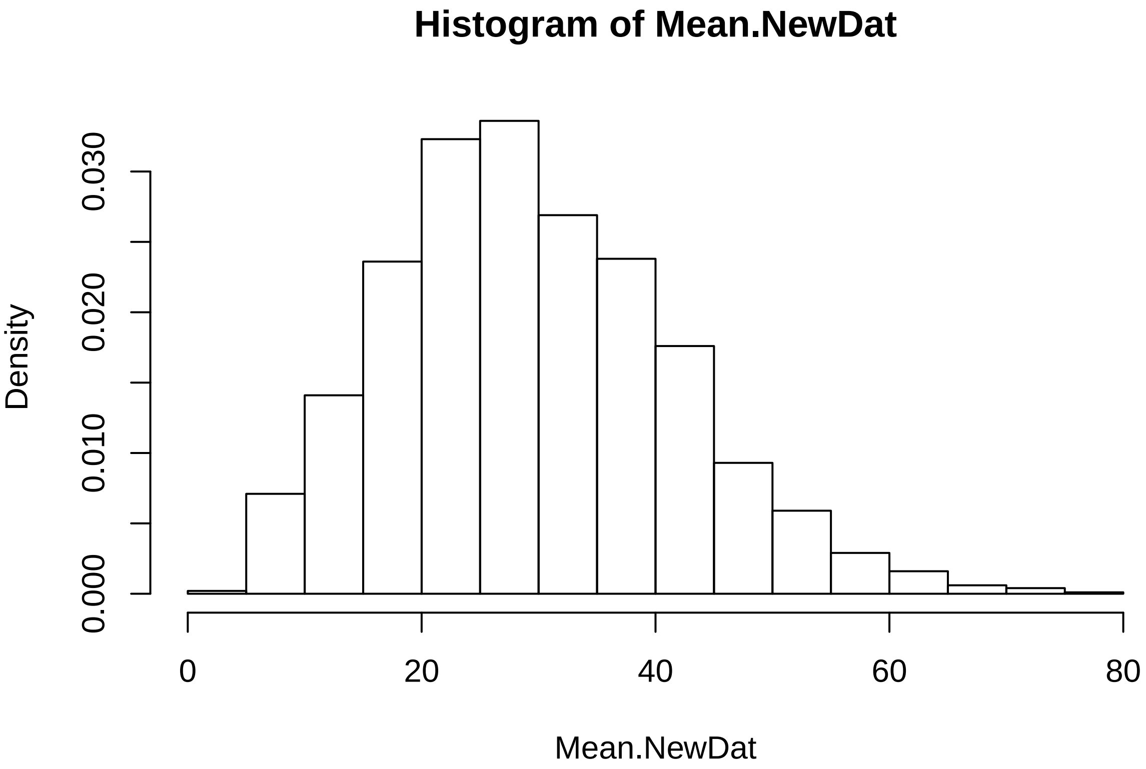
```
hist(new.data,prob=TRUE)
```



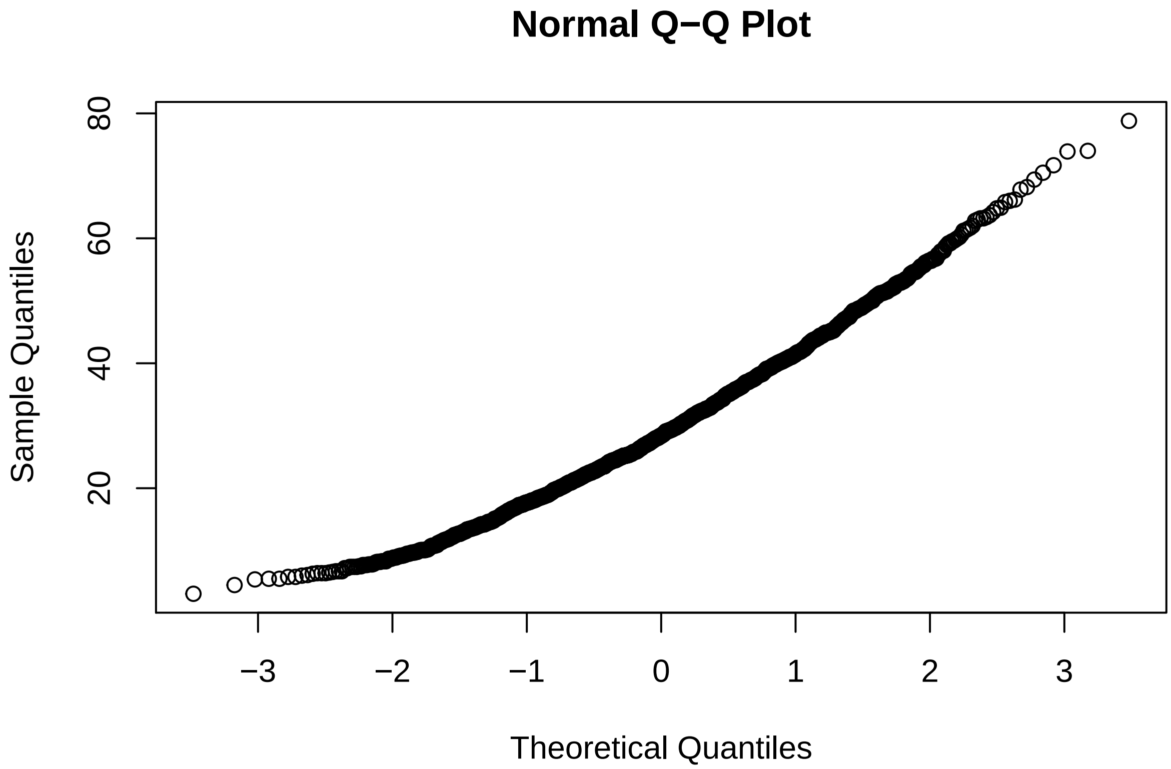
Notice, the histogram still does not show a normal distribution.

Approximating the sampling distribution of the mean and checking for normality.

```
set.seed(1)
bootsamps.2<-matrix(0,nrow=2000, ncol=length(new.data))
for(i in 1:2000){
  bootsamps.2[i,]=sample(new.data, replace=TRUE)

}
Mean.NewDat<-apply(bootsamps.2,1,mean)
## Part A
hist(Mean.NewDat, prob=TRUE)
```



```
## Part B
qqnorm(Mean.NewDat)
```



In this case, the histogram and qqplot still shows that the bootstrap distribution is approximately normal

Estimating the standard error of the mean of commute times using bootstrapping.

```
set.seed(1)
se.NewDat<-sd(Mean.NewDat)
se.NewDat
```

```
## [1] 12.07221
```

Estimating the bias of the mean of commute times using bootstrapping.

```
set.seed(1)
theta.hat.2<-mean(new.data)
bias.newDat<-mean(Mean.NewDat-theta.hat.2)
bias.newDat
```

```
## [1] 0.2902
```

Calculating 4 different 95% Confidence Intevals (Students T (no bootstrapping), Standard Normal Bootstrap CI, Percentile Bootstrap CI, Bootstrap T)

```
## Students T CI (no bootstrap)
set.seed(1)
xbar.newDat<-mean(new.data)
tcrit.newDat<-qt(.975, df=9)
seNot.newDat<-sd(new.data)/sqrt(10)
StudTCI.new<-xbar.newDat+c(-1,1)*tcrit.newDat*seNot.newDat

## Standard Normal CI
set.seed(1)
StdNormCI.new<-xbar.newDat+c(-1,1)*qnorm(.975)*se.NewDat
StdNormCI.new
```

```
## [1]  5.638902 52.961098
```

```
## Percentile
set.seed(1)
PercentCI.new<-quantile(Mean.NewDat, c(.025, .975))
PercentCI.new
```

```
##    2.5%   97.5%
##  9.0975 55.7100
```

```
## Bootstrap T
set.seed(1)
stat<-my.mean
BootTCI.new<-boot.t.ci(new.data, statistic = stat, B=2000, R=100)

## Table of Results
rbind(Table.Colname,StudTCI.new,StdNormCI.new, PercentCI.new,BootTCI.new)
```

```
##               2.5%               97.5%
## Table.Colname "Lower Bound"      "Upper Bound"
## StudTCI.new   "1.06771065786599" "57.532289342134"
## StdNormCI.new "5.63890247123647" "52.9610975287635"
## PercentCI.new "9.0975"           "55.7099999999999"
## BootTCI.new   "8.6732496135804"  "169.461343163331"
```

Comparing results from the two data sets

As you can see the estimates for standard error and bias are different between data sets. The data with smaller n has much larger standard error. In addition, one of the data sets has positive bias while the other has negative.

Another thing to notice is that the Percentile CI was the narrowest or "best" Confidence Interval in both datasets. If I had to choose only one type of CI to report it would be the Percentile Bootstrapping CI since it was the narrowest in both procedures.