Bootstrapping: An Alternative to Traditional Hypothesis Testing

MSSC 6250

Caldwell Gluesing

Tedi Cox

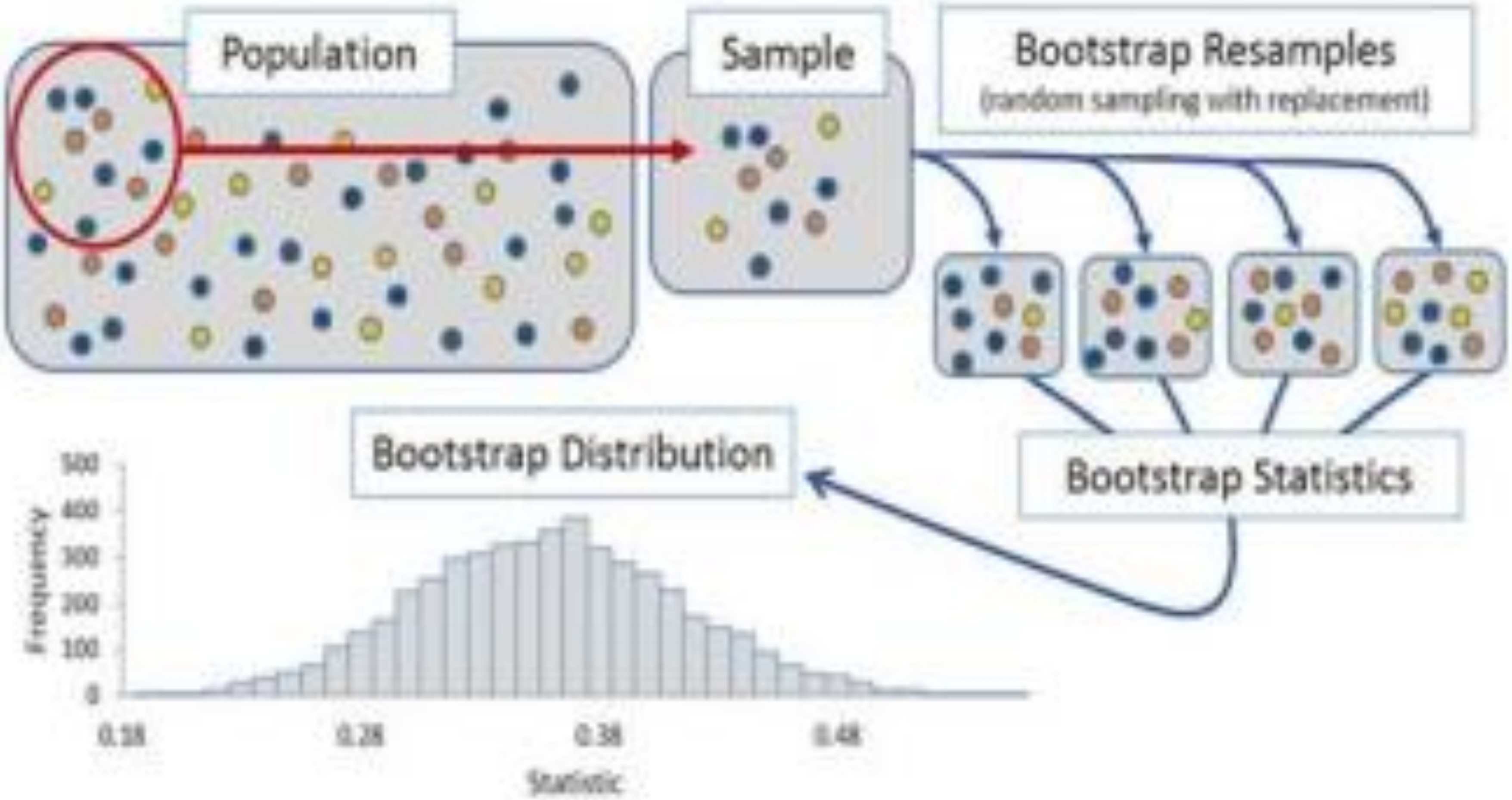


Figure 1. Summary of Bootstrapping Process

Bootstrapping (Defn 1)

Bootstrapping statistics is a form of hypothesis testing that involves resampling a single data set to create a multitude of simulated samples. Those samples are used to calculate standard errors, confidence intervals and more for hypothesis testing. This approach allows you to generate a more accurate sample from a smaller data set than the traditional method. --Trist’n Joseph

Bootstrapping (Defn 2)

Bootstrapping is a statistical procedure that resamples a single data set to create many simulated samples. This process allows for the calculation of standard errors, confidence intervals, and hypothesis testing. --Jim Frost

Questions to ponder?

What if the underlying distribution is unknown?

What if the pop. parameter is not the mean, but something more complex?

“Bootstrapping is a statistical procedure that resamples a single data set to create many simulated samples.”

Repeated Random Sampling Approach

- Draw Many Samples of size n from a population of size N
- Calculate population estimates from each of the samples
- Use distribution of population estimates to make inferences

Flaw: We may not have access to many samples from our population.

Traditional (aka Large Sample) Approach to Hypothesis Testing

- Draw One Sample of size n from a population of size N
- Calculate population estimates from the sample
- Use assumptions about population & population estimates to make inferences

When Traditional / Large Sample Approach DOESN'T work

-Insufficiently large sample size

-Ex: Central Limit Theorem (Sampling distribution of sample mean)

-Complex Statistics

-Ex: Median Absolute Deviation for regression

-Unknown population distribution

-Data with: significant outliers, extreme skew, dependencies.

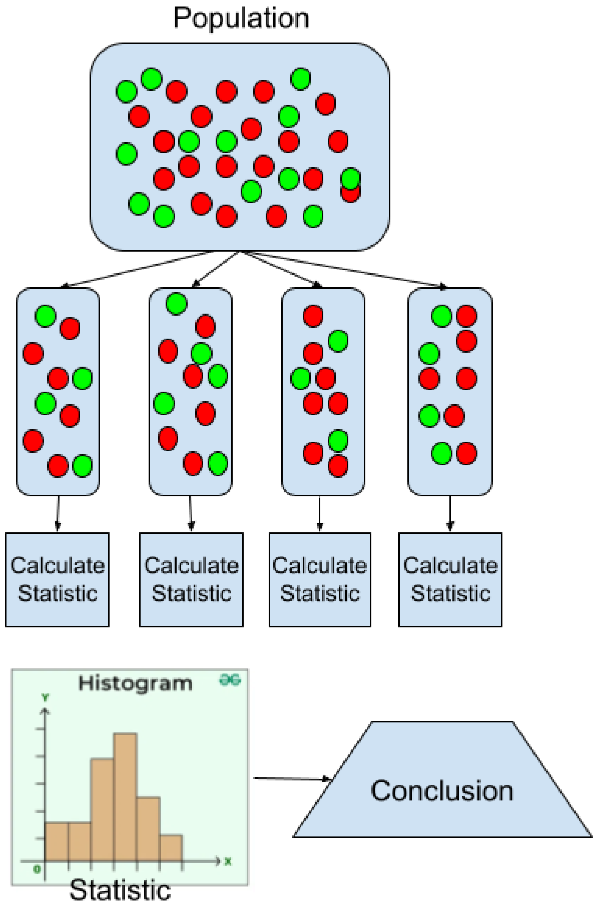
-More!

Bootstrap Sampling Approach

- Draw One sample of size n from a population of size N
- Resample from this original sample to create Many bootstrap samples
- Calculate population estimates from each of the bootstrap samples
- Use distribution of bootstrap population estimates to make inferences

Repeated Random Sampling vs Large Sample vs Bootstrap

Repeated Random Sampling



Population

Calculate Statistic

Calculate Statistic

Calculate Statistic

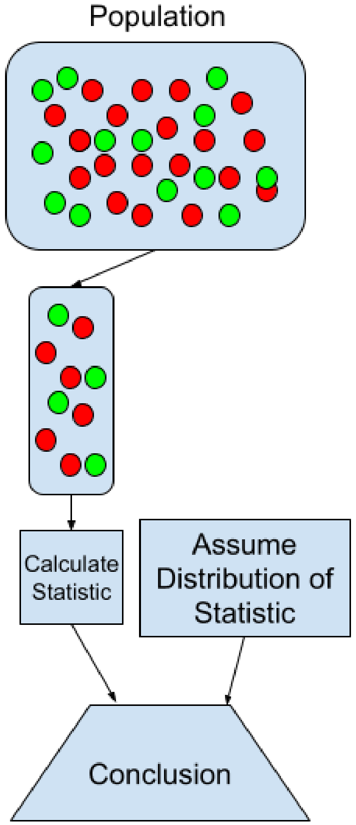
Calculate Statistic

Histogram

Statistic

Conclusion

Traditional / Large Sample



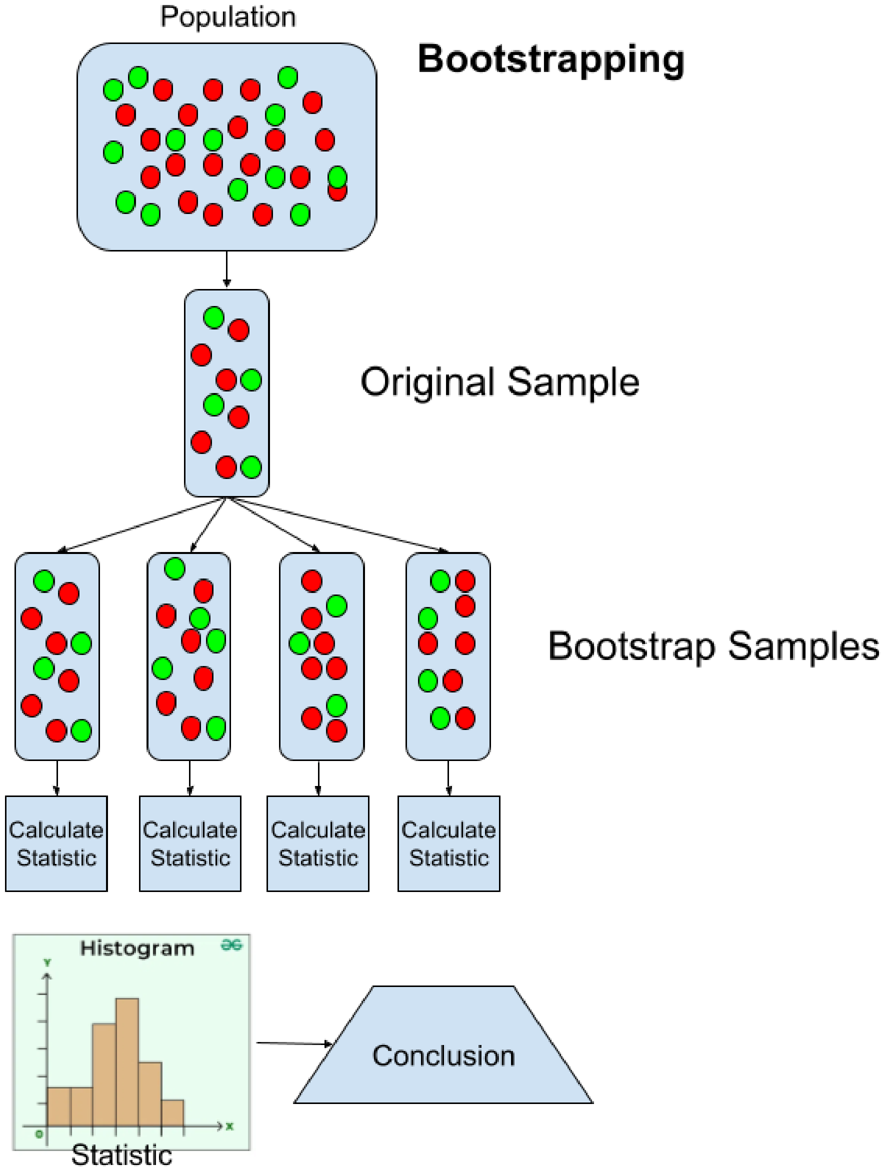
Population

Calculate Statistic

Assume Distribution of Statistic

Conclusion

Bootstrapping



Population

Original Sample

Bootstrap Samples

Calculate Statistic

Calculate Statistic

Calculate Statistic

Calculate Statistic

Histogram

Statistic

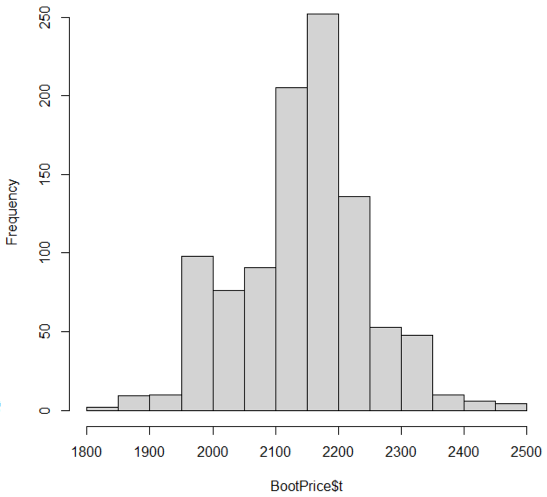
Conclusion

Bootstrapping: Basic Principles

- Each data point in the original sample has an equal chance of being chosen and resampled into a simulated sample.
- Due to replacement, a data point may be chosen more than once for the same simulated sample
- Each resampled, or simulated, sample is the same size as the original sample

Create Bootstrap Sample for the Median

1) Replace the population with the sample
2) Sample with replacement B times. (B should be large, say 1,000).
3) Computer sample medians each time, M_i.
4) Obtain the approx. distr. of the sample median.



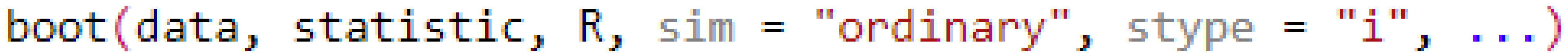
```
Price = diamonds$price #Population
n = 1000
PriceSample = sample(Price,n) #Original Sample

Median = function(Data,Indices){ #Statistic
  return(median(Data[Indices]))
}

BootPrice = boot(PriceSample,Median,R=1000) #Bootstrapping 1000 times

BootCI = boot.ci(BootPrice,type=c("norm", "basic", "perc", "bca")) #Bootstrapping CI
```

R Command: boot()



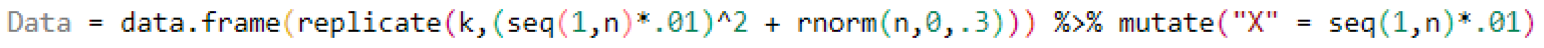
```
boot(data, statistic, R, sim = "ordinary", stype = "i", ...)
```

- data: The data set for which the bootstrap resampling is performed. Can be a vector, matrix, or data frame.
- statistic: A function that computes the statistic of interest from the data. The function must take at least two arguments: the data and an index vector of positions to sample.
- R: The number of bootstrap replicates to generate.
- sim: The method of simulation. Default is "ordinary" for bootstrap. Other options include "parametric" and "antithetic".
- stype: Specifies the type of data. "i" for indices (default), "f" for frequencies, and "w" for weights.
- ...: Additional arguments passed to the function, allowing for flexibility in computations.

R Command: boot() - Output

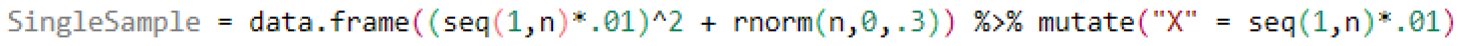
- **t0**: The original statistic computed from the data. This value is the estimate obtained from applying the statistic function to the entire dataset without resampling.
- **t**: A matrix with R rows, where R is the number of bootstrap replicates. Each row contains the statistic(s) computed from one bootstrap sample.
- **R**: The number of bootstrap replicates used in the analysis. This reflects the number of resampling iterations performed to estimate the distribution of the statistic.
- **bias**: The bias of the bootstrap estimates, calculated as the mean of the bootstrap replicates minus the original statistic (t0).
- **std.error**: The standard error of the bootstrap estimates, computed as the standard deviation of the bootstrap replicates.

Bootstrapping Approach to HW1 Q1



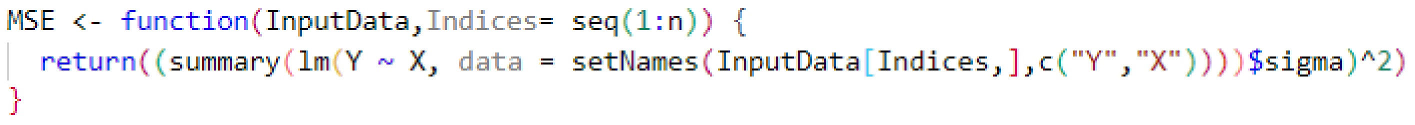
```
Data = data.frame(replicate(k,(seq(1,n)*.01)^2 + rnorm(n,0,.3))) %>% mutate("X" = seq(1,n)*.01)
```

Generate k datasets



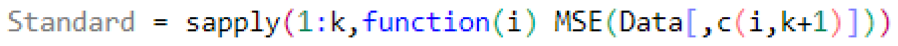
```
SingleSample = data.frame((seq(1,n)*.01)^2 + rnorm(n,0,.3)) %>% mutate("X" = seq(1,n)*.01)
```

Generate 1 dataset



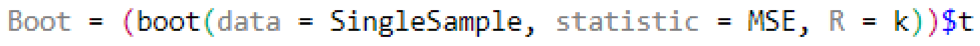
```
MSE <- function(InputData,Indices= seq(1:n)) {
  return((summary(lm(Y ~ X, data = setNames(InputData[Indices,],c("Y","X"))))$sigma)^2)
}
```

Statistic we want to calculate



```
Standard = sapply(1:k,function(i) MSE(Data[,c(i,k+1)]))
```

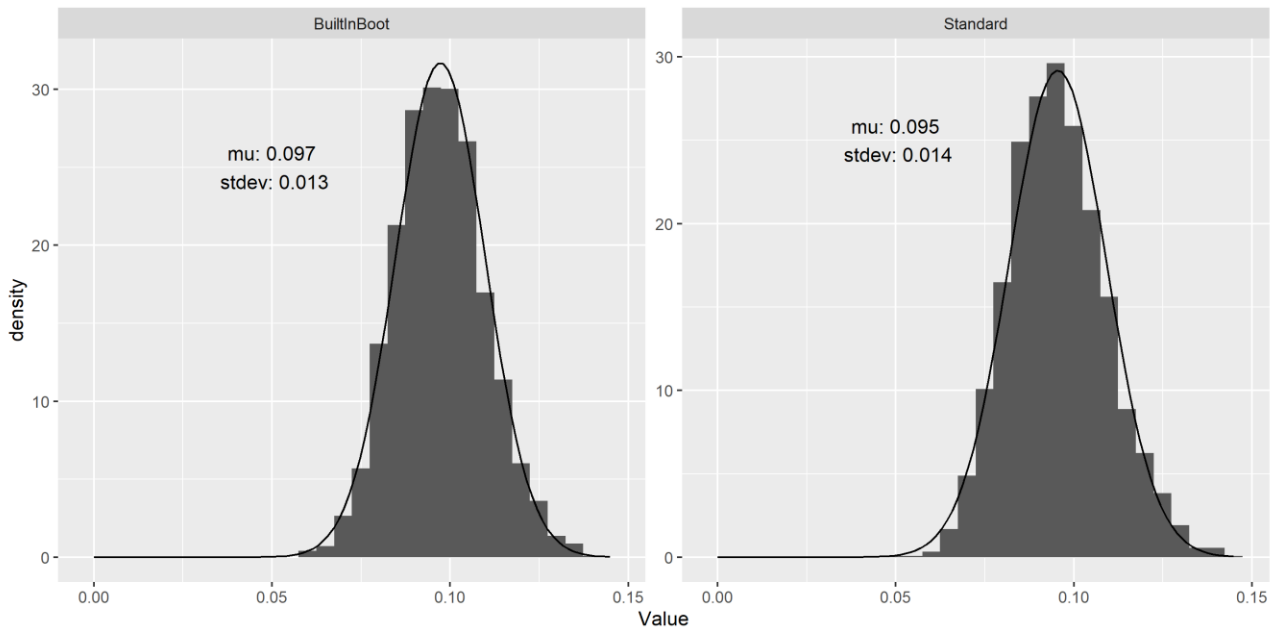
Calculate statistic on all k datasets



```
Boot = (boot(data = SingleSample, statistic = MSE, R = k))$t
```

Calculate statistic on k bootstrapped datasets

Bootstrapping Approach to HW1 Q1



Results!

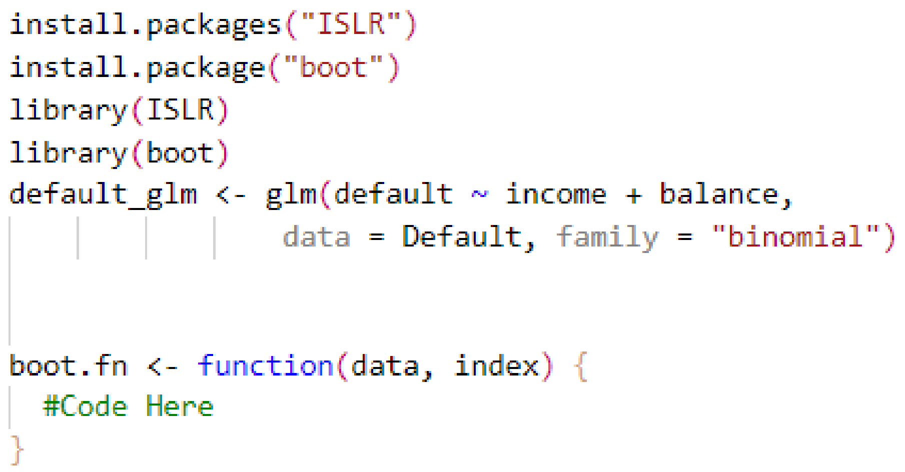
Theoretical MSE = 0.00566610 + .3^2 = 0.0956661

Model specification error Variance

Activity – ISLRv2 Chapter 5 Q6

6. Using of a logistic regression model to predict the probability of default using income and balance on the Default data set..

- (a) Using the summary() and glm() functions, determine the estimated standard errors for the coefficients associated with income and balance in a multiple logistic regression model that uses both predictors.
- (b) Write a function, boot.fn(), that takes as input the Default data set as well as an index of the observations, and that outputs the coefficient estimates for income and balance in the multiple logistic regression model.
- (c) Use the boot() function together with your boot.fn() function to estimate the standard error



```
install.packages("ISLR")
install.package("boot")
library(ISLR)
library(boot)
default_glm <- glm(default ~ income + balance,
                 data = Default, family = "binomial")


boot.fn <- function(data, index) {
  #Code Here
}
```

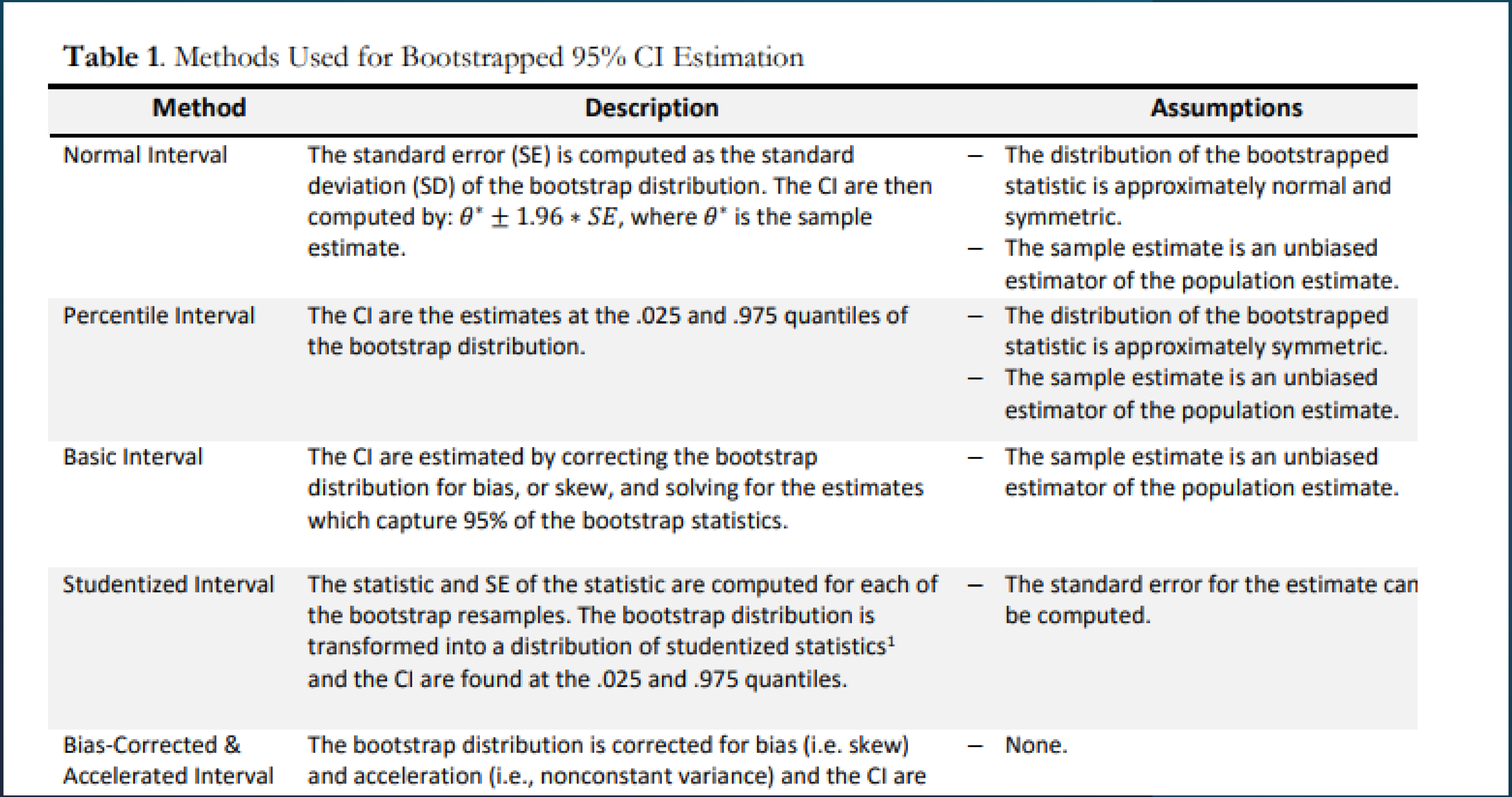
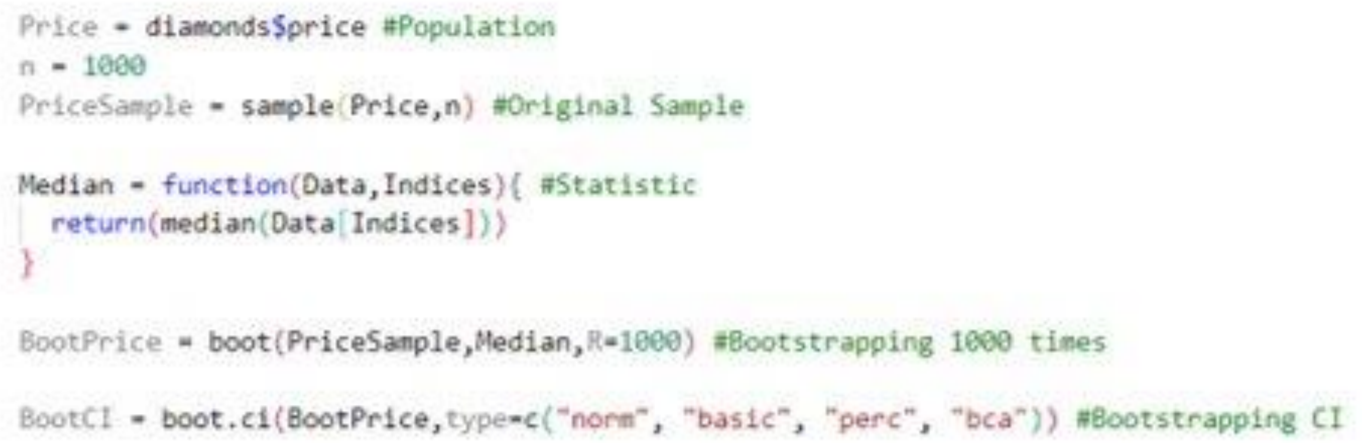


Table 1. Methods Used for Bootstrapped 95% CI Estimation

Method	Description	Assumptions
Normal Interval	The standard error (SE) is computed as the standard deviation (SD) of the bootstrap distribution. The CI are then computed by: $\theta^* \pm 1.96 * SE$, where θ^* is the sample estimate.	– The distribution of the bootstrapped statistic is approximately normal and symmetric. – The sample estimate is an unbiased estimator of the population estimate.
Percentile Interval	The CI are the estimates at the .025 and .975 quantiles of the bootstrap distribution.	– The distribution of the bootstrapped statistic is approximately symmetric. – The sample estimate is an unbiased estimator of the population estimate.
Basic Interval	The CI are estimated by correcting the bootstrap distribution for bias, or skew, and solving for the estimates which capture 95% of the bootstrap statistics.	– The sample estimate is an unbiased estimator of the population estimate.
Studentized Interval	The statistic and SE of the statistic are computed for each of the bootstrap resamples. The bootstrap distribution is transformed into a distribution of studentized statistics[1] and the CI are found at the .025 and .975 quantiles.	– The standard error for the estimate can be computed.
Bias-Corrected & Accelerated Interval	The bootstrap distribution is corrected for bias (i.e. skew) and acceleration (i.e., nonconstant variance) and the CI are	– None.

Bootstrap CI Methods

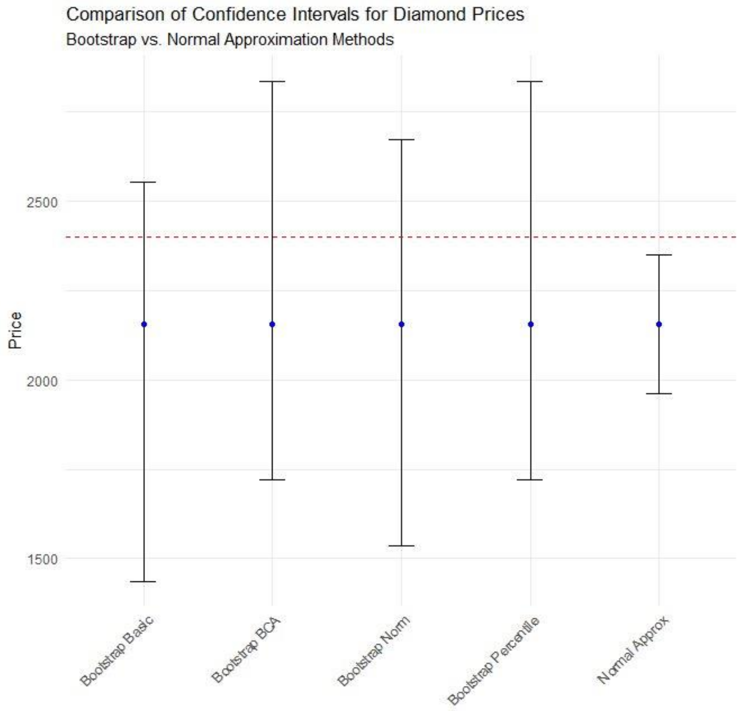


```r
Price = diamonds$price #Population
n = 1000
PriceSample = sample(Price,n) #Original Sample

Median = function(Data,Indices){ #Statistic
  return(median(Data[Indices]))
}

BootPrice = boot(PriceSample,Median,R=1000) #Bootstrapping 1000 times

BootCI = boot.ci(BootPrice,type=c("norm", "basic", "perc", "bca")) #Bootstrapping CI
```



Questions for consideration:

- Is there a formula to estimate the standard error of the statistic?
- Is the distribution symmetrical around the mean of the bootstrap resampled statistics?
- Is the distribution normal/Gaussian?
- Is the sample estimate a biased estimate of the population statistic?

Applications of bootstrap analysis for producing CIs for statistics

- Medians
- Cronbach’s alpha
- Partial eta squared
- Pseudo-R2 statistics

Limitations & Drawbacks to Bootstrapping

- Bootstrap Analysis cannot repair a fatally flawed sample, e.g. highly biased
- Different assumptions need to be checked for different methods of calculation of bootstrap-based CIs
- For better precision and narrower CIs, we need larger samples

References

- “What is Bootstrapping Statistics? By Trist’n Joseph https://towardsdatascience.com/bootstrapping-statistics-what-it-is-and-why-its-used-e2fa29577307
- “Introduction to Bootstrapping in Statistics with an Example” By Jim Frost https://statisticsbyjim.com/hypothesis-testing/bootstrapping/
- “Confidence Intervals for Effect Size: Applying Bootstrap Resampling” By Erin S. Banjanovic & Jason W. Osborne; https://scholarworks.umass.edu/cgi/viewcontent.cgi?article=1342&context=pare