

# A note on Bootstrap

gustavoc@msu.edu

10/19/2020

## Bootstrap

For a reference on this topic I suggest Chapters 10 & 11 of Computer Age Statistical Inference (Efron & Hastie, 2017).

### Context

Up until now we used inference methods that are either based on assumptions about the distribution of the data (e.g., in a linear model assuming that data are IID normal) or hold in large samples (e.g., the central limit theorem, the large sample distribution of maximum likelihood estimates).

Bootstrap is a re-sampling technique for evaluating features of the sampling distribution of an estimator (e.g., the standard error, CIs) that does not require making assumptions or residing on large-sampling arguments (although as we will see there is an implicitly large sample assumption).

### Conceptual repeated sampling

Frequentist inferences are based on the sampling distribution of the estimator.

Let  $\hat{\theta}(S_n)$  be an estimator, i.e., a function that maps from data ( $S_n$  a sample of size  $n$ , e.g.,  $S_n = \{(y_1, x_1), \dots, (y_n, x_n)\}$ ) into estimates, and let  $F$  denote the true distribution of the data. Ideally, to approximate the sampling distribution of an estimator we should:

- Draw a large number of samples  $S_{(n)j}$ , each of size  $n$ ,  $S_{(n)1}, S_{(n)2}, \dots, S_{(n)N}$ ,
- Evaluate the estimator for each samples to produce a sequence  $\hat{\theta}_1, \hat{\theta}_2, \dots, \hat{\theta}_N$ ,
- Construct an empirical distribution  $\hat{F}$  from the sequence  $\hat{\theta}_1, \dots, \hat{\theta}_N$
- Make inferences based on  $\hat{F}$ .

### Non-parametric Bootstrap

In practice we have only one sample  $S_{(n)}$ , and we do not get the benefit of drawing a large number of samples from the population (although in some contexts, e.g., surveys, historical data could be used to approximate  $\hat{F}$ ).

In Bootstrap we *pretend* that our sample  $S_n = \{(y_1, x_1), \dots, (y_n, x_n)\}$  is the population and generate *bootstrap samples* by drawing data with replacement from  $S_n$ .

This is illustrated in the following snippet which uses serum urate data from the gout data set. Our goal is to estimate the average serum urate level. We use the sample mean as our estimator and estimate the SE of the mean using the standard formula ( $SE(\bar{x}) = \sqrt{\widehat{Var}(x)/n}$ ) and using Bootstrap.

### Example 1: estimating the SE of the mean using Bootstrap

```
DATA=read.table('https://raw.githubusercontent.com/gdcl/STAT_COMP/master/goutData.txt',header=TRUE)
SU=DATA$su
n=nrow(DATA)
Estimate=mean(SU)
SEO=sqrt(var(SU)/n)

## Bootstrap
B=5000 # number of bootstrap samples
means=rep(NA,B)
for(i in 1:B){
  tmp=sample(1:n,size=n,replace=TRUE)
  bootstrap_sample=SU[tmp]
  means[i]=mean(bootstrap_sample )
}

SE.Bootstrap=sd(means)
c(SE0,SE.Bootstrap)
```

```
## [1] 0.08131497 0.08096266
```

We see that the two estimators are very close (this is expected since both are correct!).

### Example 2: Inference on the correlation coefficient

For the previous example, Bootstrap is not needed because we have a closed-form exact formula for the SE that require very minimal assumptions (random sampling). Bootstrap becomes more useful when we do not have a closed-form estimator for the SE. Although there are approximate formulas for the SE of the correlation coefficient, these formulas are only approximate and hold either in large samples or when the two RVs follow a bi-variate normal distribution.

The following example illustrates how Bootstrap approximates the sampling distribution of the correlation coefficient. To illustrate we begin by using a small sample size ( $n=50$ ). It seems that in this case bootstrap estimate smaller SE, but the two become increasingly close as sample size increases.

```
n=50
set.seed(195021)
tmp=sample(1:n,size=n)
SU=DATA$su[tmp]
AGE=DATA$age[tmp]

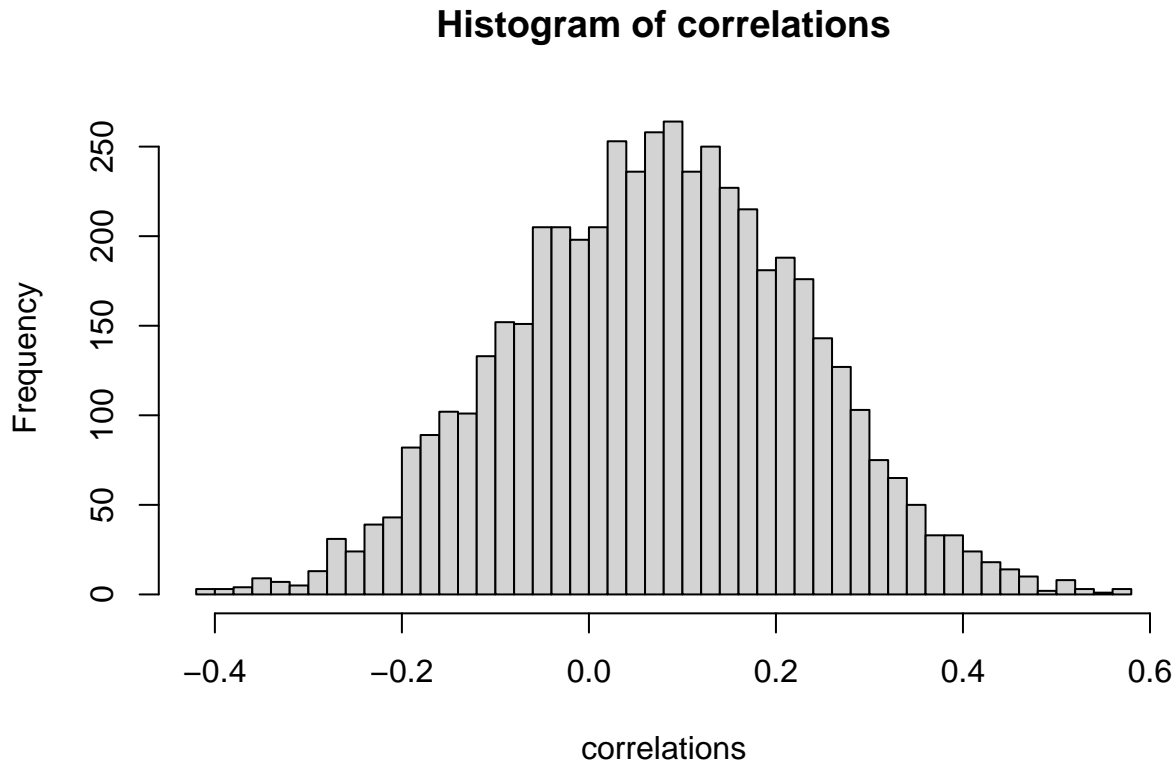
# point estimate and approximate SE
COR=cor(SU,AGE)
SEO=sqrt((1-COR^2)/(n-2))

## Bootstrap
B=5000 # number of bootstrap samples
correlations=rep(NA,B)
for(i in 1:B){
  tmp=sample(1:n,size=n,replace=TRUE)
  correlations[i]=cor(SU[tmp],AGE[tmp])
}

SE.Bootstrap=sd(correlations)
c(SE0,SE.Bootstrap)
```

```
## [1] 0.1439726 0.1549960
```

```
hist(correlations,50)
```



The following code compares the two SEs for sample equal 20,30,50,100, and 500.

```
##          n          SE0 SE.Bootstrap
## [1,]   20 0.22820792  0.21481692
## [2,]   30 0.18488187  0.17743171
## [3,]   50 0.14233557  0.13814785
## [4,]  100 0.10017789  0.09855877
## [5,]  400 0.04992928  0.04980853

## [1] 1.062337 1.041989 1.030313 1.016428 1.002424
```

## Bias

An estimator is biased if the expected value of the estimator is different than the true population parameter. In bootstrap we pretend that our sample is the population; thus we can have an approximation to the bias of the estimator by comparing the estimate that we get with the sample ( $\hat{\theta}$ ) with the average bootstrap estimate ( $\hat{\theta}_b$ ), that is:  $\hat{\theta} - \hat{\theta}_b$ . The following code evaluate potential biases for Examples 1 (sample mean) and 2 (sample correlation). The results suggest that, as expected, that the sample mean is an unbiased estimator of the population mean, but the sample correlation may be slightly upwardly biased (for correlations close to 1, the estimator may be downwardly biased).

```
DATA=read.table('https://raw.githubusercontent.com/gd1c/STAT_COMP/master/goutData.txt',header=TRUE)
n=50
set.seed(195021)
tmp=sample(1:n,size=n)
DATA=DATA[tmp,]
SU=DATA$su[tmp]
AGE=DATA$age[tmp]
```

```

# Point estimates
meanSU=mean(SU)
COR=cor(SU,AGE)

## Bootstrap
B=10000 # number of bootstrap samples
correlations=rep(NA,B)
means=correlations
for(i in 1:B){
  tmp=sample(1:n,size=n,replace=TRUE)
  correlations[i]=cor(SU[tmp],AGE[tmp])
  means[i]=mean(SU[tmp])
}

TMP=cbind('Sample Estimates'=c('mean-su'=meanSU,'cor su-age'=COR),'Average Bootstrap'=c('mean-su'=mean
TMP=cbind(TMP,'Ratio'=TMP[,2]/TMP[,1])
round(TMP,4)

##           Sample Estimates Average Bootstrap Ratio
## mean-su           5.8720           5.8702 0.9997
## cor su-age         0.0711           0.0741 1.0431

```

## Bootstrap Confidence Interval

Recall that a 95% is a decision rule that renders intervals (DATA=> CI=[Low,Up]) which, over conceptual repeated sampling, includes the true population parameter 95% of the times. There are multiple ways to use Bootstrap to estimate CIs:

- If we are willing to assume that our estimate follows a normal distribution we can use  $\hat{\theta} + / - 1.96 \times SE_{bootstrap}$ ,
- **Percentile method:** we can simply use the 0.025 and 0.975 empirical percentiles of the Bootstrap estimates to approximate the CI,
- Bias-corrected intervals.

The first approach assumes normality, thus, if normality does not hold the percentile method should be preferred. The percentile method may not be accurate if the estimate is biased, in those cases bias-corrected intervals are preferred; the function `boot.ci()` of the `boot` package can be used to compute biased-corrected CIs.

```

DATA=read.table('https://raw.githubusercontent.com/gdlc/STAT_COMP/master/goutData.txt',header=TRUE)
n=50
set.seed(195021)
tmp=sample(1:n,size=n)
DATA=DATA[tmp,]
SU=DATA$su[tmp]
AGE=DATA$age[tmp]

## Bootstrap
B=10000 # number of bootstrap samples
correlations=rep(NA,B)
for(i in 1:B){
  tmp=sample(1:n,size=n,replace=TRUE)

```

```

    correlations[i]=cor(SU[tmp],AGE[tmp])
  }

CI_1=COR+c(-1,1)*1.96*sd(correlations)
CI_2=quantile(correlations,p=c(.025,.975))

round(CI_1,3)

## [1] -0.233  0.376

round(CI_2,3)

##    2.5%  97.5%
## -0.232  0.374

```