Applied Bootstrap

E. F. Haghish

3 May 2016

NOTE: **Read the article by Efron entitled:** *Bootstrap methods- another look at the jackknife.pdf*
You can find it on GitHub under **ST516/Articles/Bootstrap/**

Bootstrap

- It was developed by **Efron** in 1979
- It was inspired by **Jacknife** method
- Bootstrap generates random sample from empirical distribution
- Bootstrap is sampling with replacement
- It is often used for estimating **standard error** and **bias**

Bootstrap methods are nonparametric Monte Carlo methods that estimate the population distribution by **resampling** from an observed sample. Naturally, this method is used when the population mean μ is unknown. The resampling method allows us to *estimate population characteristics* and make inference about them.

- Note that bootstrap can also be done from a probability distribution, known as *parametric bootstrap* which does not concern us here. We focus on the nonparametric bootsrap.

Obtaining distribution function from the sample

The bootstrap estimates of a sampling distribution are analogous to **density estimation**. We would like to discover the distribution density of the population using a finite available sample.

- we construct a histogram of a sample to obtain an estimate of the shape of the sample
- when we sample from a distribution, observations with higher density are more likely to be selected
- The enpirical distribution function is in fact the Cumulative Distribution Function (CDF) of the sample
- Resampling from the sample is similar to resampling from the **ecdf**

P. 145

Remember `ecdf` function in R.

```
plot(ecdf(rnorm(100)))
```

ecdf

$$F_e = \frac{\sum_1^i : X_i \leq x}{n}$$

Bootstrap estimate of Standard Error

- In the previous session I discussed that when we have multiple samples, the standard error is the square root of variance of samples mean:

$$E\left[(\overline{X} - \theta)^2\right] = Var(\overline{X}) = \frac{\sigma^2}{n}$$

$$SE = \frac{\sigma}{\sqrt{n}}$$

The bootstrap estimate of Standard error of an estimator $\widehat{\theta}$ is the sample standard deviation of the bootstrap replicates.
n: is number of replicates

$$\widehat{se} = \sqrt{\frac{1}{N-1}\sum_{i=1}^{n}(\widehat{x}_i - \overline{\widehat{x}})^2}$$

Example 1

The `law` data is from **Efron & Tibshirani 1993** *An Introduction to Bootstrap* - LSAT: Average score on Law school administration test score - GPA: average undergraduate grade-point average for 15 law schools
The `law82` data set is a random sample of 82 law schools

```
install.packages("bootstrap")
library(bootstrap)
# View(law) #to view the data in RStudio
```

Use bootstrap to estimate the standard error of the correlation between the LSAT and GPA variables

```r
#correlation
cor(law$LSAT,law$GPA)
```

```
## [1] 0.7763745
```

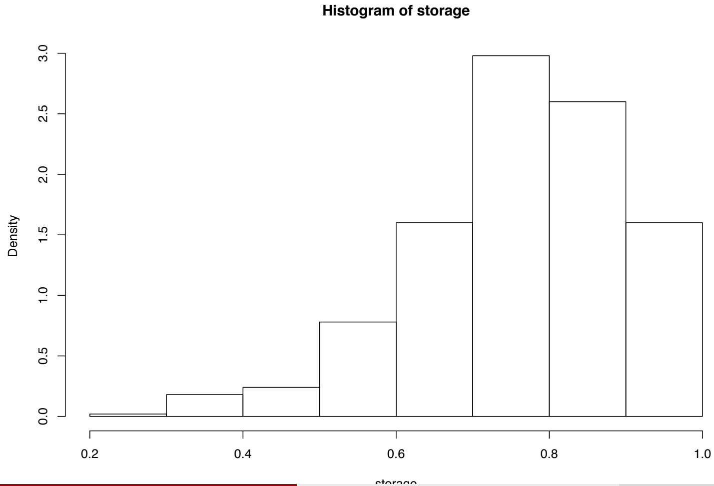
1. We should take bootstrap samples from our sample variables
2. For each pair of samples, calculate the correlation
3. We will have a vector that stores all the obtained correlations
4. We calculate the **Standard Deviation** of the vector, which results in **standard error** of the bootstrap correlation

```r
set.seed(516)
N <- 500                  #number of replicates
n <- nrow(law)            #sample size (number of rows)
storage <- numeric(N) #storage variable
for (i in 1:N) {
    j <- sample(1:n, size = n, replace = TRUE) #random indice
    LSAT    <- law$LSAT[j]
    GPA     <- law$GPA[j]
    storage[i] <- cor(LSAT, GPA)
}
print(se <- sd(storage))
```

```
## [1] 0.1342233
```

Histogram based on density (probability)

```
hist(storage, probability = TRUE) #instead of frequency
```



storage

The storage variable includes a vector of the estimates that were generated from each bootstrapped sample. The bootstrapped estimate is the **mean** of this vector:

```
head(storage)
```

```
## [1] 0.6794599 0.6753038 0.8558274 0.8664717 0.7379599 0.519
```

```
mean(storage)
```

```
## [1] 0.7638062
```

Example 2: Using the “boot” package

The `boot` package can make running bootstrap much more convenient. However, for this you will have to write a **statistics** function and pass it to `boot` function.

```
install.packages("boot")
library(boot)
```

`statistics` function

The statistics function is the function that is used for estimating the parameter of interest in each sample of the bootstrap. The `statistics` function should take 2 arguments, respectively:

```
1. sample data `x`
2. vector of indices, required for selecting the observations
```

I will use the `boot` function to repeat the same computation

```r
set.seed(516)
correlation <- function(x,i) {
    #1 and #2 are column number for law dataset
    cor(x[i,1], x[i,2])
}
c <- boot(data=law, statistic = correlation, R = 50) #repeat 
c
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = law, statistic = correlation, R = 50)
##
##
## Bootstrap Statistics :
##      original       bias    std. error
```

The “original” value is the **mean** of the $\widehat{\theta}$ or observed value. The **SD** of ot will reveal the bootstrap **SE**.

```
sd(c$t) #obtaining value t from the list
```

```
## [1] 0.1252061
```

Bootstrap estimation of Bias

Biase is the difference between the value of estimator and the parameter

$$bias(\widehat{\theta}) = E\left[\widehat{\theta} - \theta\right] = E\left[\widehat{\theta}\right] - \theta$$

Example 3

Let's repeat example 1 and **estimate the bias of sample correlation**

```
set.seed(516)
theta.hat <- cor(law$LSAT, law$GPA)
N <- 500
n <- nrow(law)
storage <- numeric(N)
for (i in 1:N) {
    j <- sample(1:n, size=n, replace = TRUE)
    LSAT <- law$LSAT[j]
    GPA  <- law$GPA[j]
    storage[i] <- cor(LSAT,GPA)
}
theta.hat.boot <- mean(storage)
bias <- theta.hat.boot - theta.hat
bias
```

Standard Normal Bootstrap Confidence Interval

- The simplest form of Confidence Interval
- Relies on **Central Limit Theorem** so it requiers large sample to be effective
- It assumes $\widehat{\theta}$ is unbiased
- It assumes normal distribution
- Then θ is in the Z interval

$$\widehat{\theta} \pm z_{\frac{\alpha}{2}} SE(\widehat{\theta})$$

where:

- α is the p-value i.e. 0.05 or lower (or anything you like)
- $z_{\frac{\alpha}{2}}$ = inverse of cdf for (1- $\frac{\alpha}{2}$)
- $= \Phi^{-1}(1 - \frac{\alpha}{2})$
- $= \texttt{qnorm}(1 - \frac{\alpha}{2})$

Lower CI: