Non-parametric Statistics: Notes 8
Nonparametric Bootstrap Methods

Gregory Matthews [1]

[1] Department of Mathematics and Statistics
Loyola University Chicago

Fall 2014

Outline

- In statistics, we use data to estimate population parameters.
- When we estimate a parameters, there is inherently error involved.
- So we are often interested in how accurate our estimator is to the population value.

- For instance, we can estimate the population parameter μ with $\bar{X}$
- But how accurate is that estimate?
- What we are interested in is the $Var(\bar{X})$ a measure of variability for out estimate.
- $Var(\bar{X}) = \frac{\sigma^2}{n}$ where σ^2 is the variance of X and n is the sample size.

- If we replace σ^2 with s^2 and invoke the Central Limit Theorem (CLT), we get a 95% confidence interval of the form: $\bar{X} \pm 1.96\frac{s}{\sqrt{n}}$
- But what about situations where the CLT does not help us?
- We can use a technique called bootstrapping.
- The bootstrap was developed by Bradley Efron.

- Sidenote: Efron's Dice.
- Consider 4 dice with the following sides:
  - A: 4,4,4,4,0,0
  - B: 3,3,3,3,3,3
  - C: 6,6,2,2,2,2
  - D: 5,5,5,1,1,1
- Consider a game where the object is to roll the highest number. You choose a die and then I choose a die. We both roll and the higher number wins. What die would you choose to roll?

- No matter what die you choose, I can choose a die that will win $\frac{2}{3}$ of the time.
- In fact, $P(A > B) = P(B > C) = P(C > D) = P(D > A) = \frac{2}{3}$
- Incredible.

- Ok. Back to the bootstrap.
- First let's talk about mean squared error (MSE) and margin of error (MOE).

- Let θ be some population parameter and $\hat{\theta}$ be a statistical estimator of θ based on a sample of size n.
- Then MSE $= E(\hat{\theta} - \theta)^2$
- This is the average squared deviation from the estimate to the population parameter.
- For the sample mean, MSE $= \frac{\sigma^2}{n}$

- MSE is important because of the result of the Chebyshev-Markov inequality.

$$P(|\hat{\theta} - \theta| \leq k\sqrt{MSE}) \geq 1 - \frac{1}{k^2}$$

- When $k = 2$, then $1 - \frac{1}{k^2} = 1 - \frac{1}{4} = \frac{3}{4}$.
- Often it is much more than this.
- The quantity $2\sqrt{MSE}$ or $(1.96\sqrt{MSE})$ is referred to as the margin of error.

- Often there is not a formula for explicitly calculating the MSE.
- If we had some way of sampling from the population we could estimate the MSE by repeated sampling from the distribution and calculating the quantity:
- However, the population distribution is usually unknown so we need another approach.

$$\hat{MSE} = \frac{1}{nsim} \Sigma_{i=1}^{nsim} (\hat{\theta}_i - \theta)^2$$

- What can we do? BOOTSTRAP!
- The bootstrap procedure is based on resampling.
- In the absense of the the population, we use the data as a substitute.
- The key to simulating sampling from an infinite population is to sample WITH replacement.

Bootstrapping the MSE

1. Computer $\hat{\theta}$ from the original data.
2. Take *nsim* bootstrap samples of size n from the data. Let *nsim* denote the number of boot strap samples. Typically, $nsim \geq 1000$.
3. Compute $\hat{\theta}_{b,i}$, the estimate of θ obtained from the i-th bootstrap sample.
4. Obtain the bootstrap MSE as

$$\hat{MSE} = \frac{1}{nsim} \Sigma_{i=1}^{nsim} (\hat{\theta}_{b,i} - \hat{\theta})^2$$

```
#Let's bootstrap!
set.seed(1234)
n<-15
#Randomly sample from a population
x<-rnorm(n,20,5)
x

##  [1] 13.965 21.387 25.422  8.272 22.146 22.530 17.126 17
## [11] 17.614 15.008 16.119 20.322 24.797

(thetaHat<-mean(x))

## [1] 18.31

#true MSE
25/n

## [1] 1.667
```

```r
nsim<-1000
thetaBoots<-rep(NA,nsim)
#replace = TRUE is key!
for (i in 1:nsim){
  bootsSample<-x[sample(1:n,n,replace=TRUE)]
  thetaBoots[i]<-mean(bootsSample)
}
#Bootstrap estimate of MSE
mean((thetaBoots-thetaHat)^2)

## [1] 1.283
```

```r
#Let's bootstrap!
set.seed(12345)
n<-200
#Randomly sample from a population
x<-rnorm(n,20,5)
(thetaHat<-mean(x))

## [1] 20.73

#true MSE
25/n

## [1] 0.125
```

```r
nsim<-1000
thetaBoots<-rep(NA,nsim)
#replace = TRUE is key!
for (i in 1:nsim){
  bootsSample<-x[sample(1:n,n,replace=TRUE)]
  thetaBoots[i]<-mean(bootsSample)
}
#Bootstrap estimate of MSE
mean((thetaBoots-thetaHat)^2)

## [1] 0.1366
```

```r
#Let's bootstrap!
set.seed(123456)
#No reason we have to only use the mean
n<-100
#Randomly sample from a population
x<-rnorm(n,20,5)
thetaHat<-var(x)
```

```r
#calculate the true MSE
nsim<-1000
getReal<-rep(NA,nsim)
#replace = TRUE is key!
for (i in 1:nsim){
  samp<-rnorm(n,20,5)
  getReal[i]<-var(samp)
}
#Bootstrap estimate of MSE of sigma^2 hat
mean((getReal-25)^2)

## [1] 12.86
```

```
nsim<-1000
thetaBoots<-rep(NA,nsim)
#replace = TRUE is key!
for (i in 1:nsim){
  bootsSample<-x[sample(1:n,n,replace=TRUE)]
  thetaBoots[i]<-var(bootsSample)
}
#Bootstrap estimate of MSE of sigma^2 hat
mean((thetaBoots-thetaHat)^2)

## [1] 10.89
```

Bootstrap Variance and Bias

- Recall that bias is the difference between the expected value of an estimate and the quantity being estimated.
- And MSE can be expressed as the variance plus the bias squared.

$$B = E[\hat{\theta}] - \theta$$

$$MSE = var + B^2$$

Bootstrap Variance and Bias

- Like MSE, we can obtain bootstrap estimates of the varie and bias.

$$\hat{E} = \frac{1}{nsim} \Sigma_{i=1}^{nsim} \hat{\theta}_{b,i}$$

$$\hat{B} = \hat{E} - \hat{\theta}$$

$$var = \frac{1}{nsim} \Sigma_{i=1}^{nsim} (\hat{\theta}_{b,i} - \hat{E})^2$$

How many samples

- Booth and Sarkar (1998) recommend at least 800 samples for estimating the variance of $\hat{\theta}$.
- However, with improvements in computing since 1998, the number of samples can be pretty much as large as we need it to be (5000 or 10000 or more) without a significant computing burden.

Note: Parametric Bootstrap

- Non-parametric bootstrap makes no assumptions about the distribution of the data.
- A parametric distribution makes assumptions about the data.
- For instance, we could assume that the data is normal and estimate the parameters of this distribution with the data.
- Bootstrap samples would then be drawn from this normal distribution.

Bootstrap Intervals

- If the data comes from a normal distribution we can do the following.
- Define the *pivot* quantity $t = \frac{\bar{X}-\mu}{\frac{S}{\sqrt{n}}}$ which has a t-distribution with $n-1$ degrees of freedom.
- Then $P(-t_{.975} \leq t \leq t_{.975}) = 0.95$.
- Solving for μ gives us the familiar 95% confidence interval: $\bar{X} - t_{.975}\frac{S}{\sqrt{n}} \leq \mu \leq \bar{X} + t_{.975}$

Bootstrap Intervals

- If we throw out the assumption of normality we can still use t.
- Define the *pivot* quantity $t = \frac{\bar{X}-\mu}{\frac{S}{\sqrt{n}}}$ ~~which has a t-distribution with $n-1$ degrees of freedom.~~
- Then $P(t_{.025} \leq t \leq t_{.975}) = 0.95$.
- Solving for μ gives us the familiar 95% confidence interval: $\bar{X} - t_{.975}\frac{S}{\sqrt{n}} \leq \mu \leq \bar{X} - t_{.025}$
- (Of course we could adjust this procedure for any desired confidence limit.)

Steps for obtaining a Bootstrap interval for μ

1. Compute the mean $\bar{X}$ and the standard deviation S of the original data.
2. Obtain a bootstrap sample of size n from the data. Compute the mean $\bar{X}_b$ and the standard deviation S_b of the bootstrap sample, and compute the t-pivot quantity: $t_b = \frac{\bar{X}_b - \bar{X}}{\frac{S_b}{\sqrt{n}}}$.
3. Repeat step 2 a number of times - say 1000 or more - to obtain a boostrap distribution of the t_b's.
4. For a 95% confidence interval, let $t_{b,0.025}$ and $t_{b,0.975}$ be the 2.5th and the 97.5th percentiles of the bootstrap distribution. The 95% bootstrap interval is then
$\bar{X} - t_{b,.975}\frac{S}{\sqrt{n}} \leq \mu \leq \bar{X} - t_{b,.025}$

```r
#Speal Length for 50 setosas
(x<-iris$Sepal.Length[iris$Species=="setosa"])

##  [1] 5.1 4.9 4.7 4.6 5.0 5.4 4.6 5.0 4.4 4.9 5.4 4.8 4.8
## [18] 5.1 5.7 5.1 5.4 5.1 4.6 5.1 4.8 5.0 5.0 5.2 5.2 4.7
## [35] 4.9 5.0 5.5 4.9 4.4 5.1 5.0 4.5 4.4 5.0 5.1 4.8 5.1
```

```
(xbar<-mean(x))

## [1] 5.006

(s<-sd(x))

## [1] 0.3525

n<-length(x)
#t-interval
c(xbar-qt(0.975,n-1)*s/sqrt(n),xbar+qt(0.975,n-1)*s/sqrt(n

## [1] 4.906 5.106
```

```r
#Now we need to bootstrap to find the percentile of the t p
set.seed(1234)
nsim<-1000
tBoot<-rep(NA,nsim)
for (i in 1:nsim){
xBoot<-sample(x,n,replace=TRUE)
tBoot[i]<-(mean(xBoot)-xbar)/(sd(xBoot)/sqrt(n))
}
(tq<-(quantile(tBoot,c(0.025,0.975))))

##   2.5%  97.5%
## -1.973  2.044
```

```
#Bootstrap interval
c(xbar-tq[2]*s/sqrt(n),xbar-tq[1]*s/sqrt(n))

## 97.5%  2.5%
## 4.904 5.104

#t-interval
c(xbar-qt(0.975,n-1)*s/sqrt(n),xbar+qt(0.975,n-1)*s/sqrt(n)

## [1] 4.906 5.106
```

```r
#20 observations from a non-normal distribution
set.seed(12345)
(x<-rchisq(20,1))

##  [1] 5.842e-01 3.622e-06 5.997e-01 4.541e-01 4.335e+00
##  [8] 1.005e-04 1.303e+00 3.666e-01 5.036e-02 2.893e-01
## [15] 1.809e-02 1.876e-01 4.216e+00 3.899e-01 6.527e-03
```

```r
(xbar<-mean(x))

## [1] 0.7806

(s<-sd(x))

## [1] 1.242

n<-length(x)
#t-interval
#Assumtpion of normality is wrong here
c(xbar-qt(0.975,n-1)*s/sqrt(n),
  xbar+qt(0.975,n-1)*s/sqrt(n))

## [1] 0.1994 1.3619
```

```
#Now we need to bootstrap to find the percentile of the t
set.seed(12345)
nsim<-1000
tBoot<-rep(NA,nsim)
for (i in 1:nsim){
xBoot<-sample(x,n,replace=TRUE)
tBoot[i]<-(mean(xBoot)-xbar)/(sd(xBoot)/sqrt(n))
}
(tq<-(quantile(tBoot,c(0.025,0.975))))

##   2.5%  97.5%
## -6.476  1.538
```

```r
#Bootstrap interval
c(xbar-tq[2]*s/sqrt(n),xbar-tq[1]*s/sqrt(n))

##  97.5%   2.5%
## 0.3535 2.5789

#t-interval
c(xbar-qt(0.975,n-1)*s/sqrt(n),
  xbar+qt(0.975,n-1)*s/sqrt(n))

## [1] 0.1994 1.3619
```

Percenitle method

Percentile and residual methods

- Draw a specified number of bootstrap samples of size n from the data, and for each bootstrap sample compute the estimate $\hat{\theta}_b$ of θ.
- For a 95% confidence interval for θ, find the 2.5th and 97.5th percentiles of the bootstrap distribution. The percentile-method boostrap 95% confidence interval is

$$\hat{\theta}_{b,0.025} < \theta \leq \hat{\theta}_{b,0.025}$$

- Make the appropriate modifications for other levels of confidence.

Residual method

- Compute $\hat{\theta}$ from the data.
- Draw a bootstrap sample of size n from the data. Compute $\hat{\theta}_b$ and the residual $e_b = \hat{\theta}_b - \hat{\theta}$.
- Repeat step 2 a specified number of times to obtain a boostrap distribution of the e_b's.
- For a 95% confidence interval, obtain the 2.5th and 97.5th percentile $e_{b,0.025}$ and $e_{b,.975}$, of the bootstrap distribution. The confidence interval for θ is

$$\hat{\theta} - e_{b,.975} \leq \theta \leq \hat{\theta} - e_{b,.025}$$

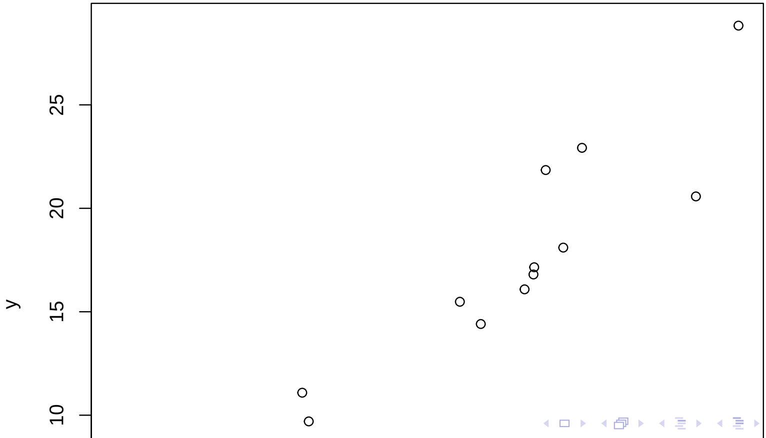
Bootstrap for correlation

- What if we are interested is sampling from a bivariate distribution?
- Why would this be usedul?
- What if we wanted to bootstrap the correlation coefficient?

Bootstrap for correlation

1. Using bivariate sampling, sample *nsim* boot strap samples from the data.
2. For each bootstrap sample, compute the Pearson correltion coefficient, ρ.
3. Construct a confidence interval using the percentile method.

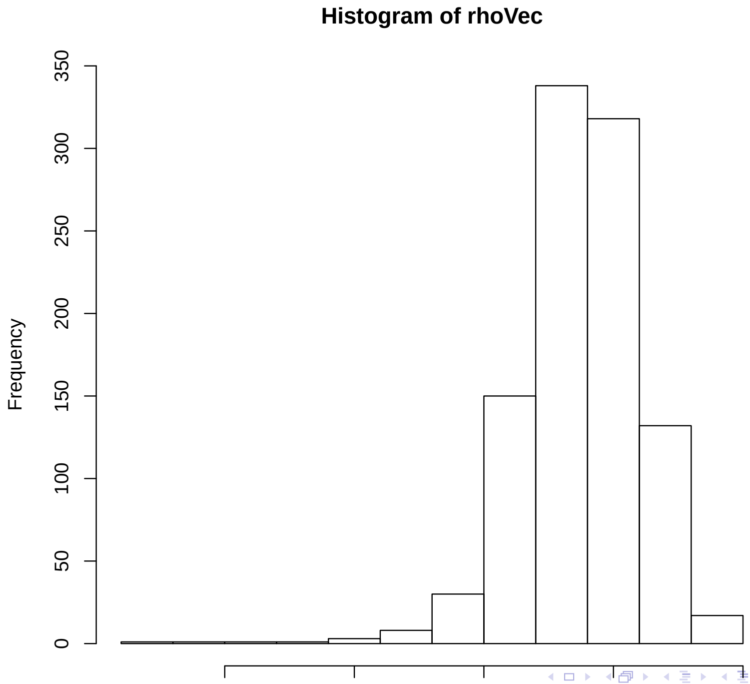
```r
set.seed(1234)
x<-runif(15,0,10)
y<-3*x+rnorm(15,0,3)
plot(x,y)
```



```r
cbind(x,y)[1:10,]

##              x      y
##  [1,] 1.13703  6.361
##  [2,] 6.22299 16.802
##  [3,] 6.09275 16.084
##  [4,] 6.23379 17.151
##  [5,] 8.60915 20.575
##  [6,] 6.40311 21.850
##  [7,] 0.09496  4.395
##  [8,] 2.32551  1.915
##  [9,] 6.66084 18.100
## [10,] 5.14251 15.482
```

```r
dat<-data.frame(x,y)
cor(dat)

##          x      y
## x 1.0000 0.9324
## y 0.9324 1.0000

rho<-cor(dat)[1,2]
```

```r
nsim<-1000
rhoVec<-rep(NA,nsim)
n<-dim(dat)[1]
for (i in 1:nsim){
  datBoots<-dat[sample(1:n,n,replace=TRUE),]
  rhoVec[i]<-cor(datBoots)[1,2]
}
```

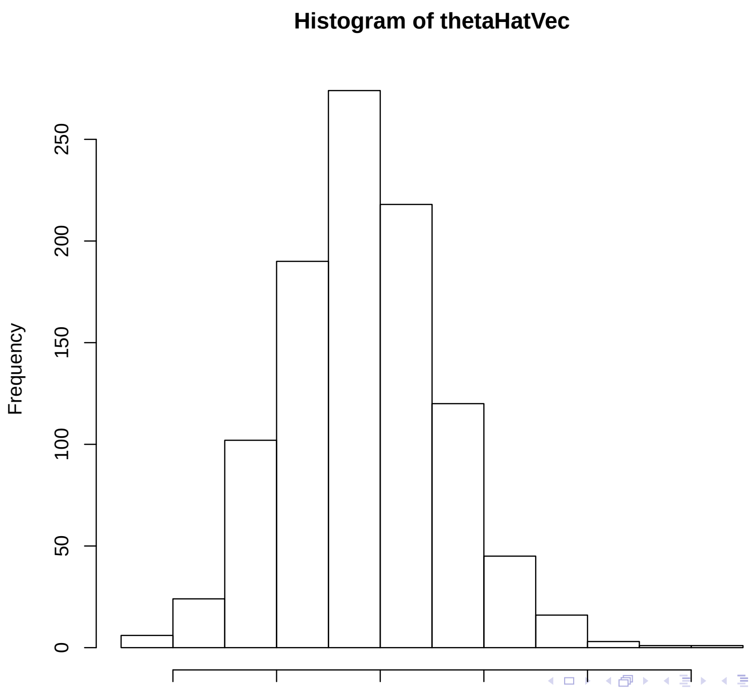


Histogram of rhoVec
Frequency
0
50
100
150
200
250
300
350

```
#percentile method
#95% CI for rho
quantile(rhoVec,c(0.025,.975))

##   2.5%  97.5%
## 0.8930 0.9763
```

```
#Repeat the same thing, but for the ratio of means
dat<-data.frame(x,y)
means<-apply(dat,2,mean)
thetaHat<-means[2]/means[1]
```

```
nsim<-1000
thetaHatVec<-rep(NA,nsim)
n<-dim(dat)[1]
for (i in 1:nsim){
  datBoots<-dat[sample(1:n,n,replace=TRUE),]
  means<-apply(datBoots,2,mean)
  thetaHatVec[i]<-means[2]/means[1]
}
```


Histogram of thetaHatVec
Frequency
0
50
100
150
200
250

```r
#percentile method
#95% CI for rho
quantile(thetaHatVec,c(0.025,.975))

##  2.5% 97.5%
## 2.688 3.281
```