The Bootstrap Method

Section 5.2

Dr. Cathy Poliak, cpoliak@uh.edu

University of Houston

Resampling Methods

- **Resampling methods** involve repeatedly drawing samples from a training set and refitting a model of interest on each sample in order to obtain additional information about the fitted model.
- Could potentially be computationally expensive, because they involve fitting the same statistical method multiple times using different subsets of the training data.
- Two most commonly used resampling methods:
  - Cross-validation - can be used to estimate the test error associated with a given statistical learning method in order to evaluate its performance.
  - Bootstrap - can be used to provide a measure of accuracy of a parameter estimate or of a given statistical learning method.

The Bootsrap Method

- The **bootstrap** is a widely applicable and extremely powerful statistical tool that can be used to quantify the uncertainty associated with a given estimator or statistical learning method.
- The machine learning techniques that use the bootstrap is the *tree*-based models: Bagging, Random Forest, etc.
- The power of the bootstrap lies in the fact that it can be easily applied to a wide range of statistical learning methods, including some for which a measure of variability is otherwise difficult to obtain and is not automatically output by statistical software.

Idea of the Bootsrap

- Resample from the original data - either directly or via a fitted model - to create data sets, from which the variability of the quantities of interest can be assessed with out long-winded and error-prone analytically calculations.
- This approach involves repeating the original data analysis procedure with many replicate sets of data.
- The central goal is to obtain reliable standard errors, confidence intervals, and other measures of uncertainty for a wide range of problems.
- This approach can be applied in simple problems to check the adequacy of standard measures of uncertainty, to relax assumptions, and to give quick approximate solutions.
- The basic idea of bootstrap is to make inference about an estimate (such as the sample mean or sample coefficients $\hat{\beta}_j$) for a population parameter θ (such as the population mean or coefficients β_j) on sample data.

Steps for Bootstrap

1. Get a sample from a population with sample size n.

Steps for Bootstrap

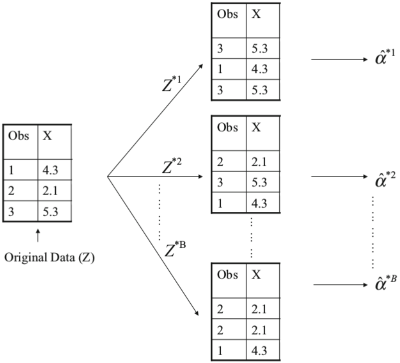
1. Get a sample from a population with sample size n.
2. Draw a sample from the original sample data **with replacement** with size n, and replicate B times, each re-sampled sample is a called a **Bootstrap Sample**, and there will be totally B Bootstrap Samples.

Steps for Bootstrap

1. Get a sample from a population with sample size n.
2. Draw a sample from the original sample data **with replacement** with size n, and replicate B times, each re-sampled sample is a called a **Bootstrap Sample**, and there will be totally B Bootstrap Samples.
3. Evaluate the statistic of θ for each Bootstrap Sample, and there will be a total of B estimates of θ.

Steps for Bootstrap

1. Get a sample from a population with sample size n.
2. Draw a sample from the original sample data **with replacement** with size n, and replicate B times, each re-sampled sample is a called a **Bootstrap Sample**, and there will be totally B Bootstrap Samples.
3. Evaluate the statistic of θ for each Bootstrap Sample, and there will be a total of B estimates of θ.
4. Construct a **sampling distribution** with these B Bootstrap statistics and use it to make further statistical inference, such as:
    - Estimating the standard error of the statistic for θ.
    - Obtaining a confidence interval for θ.



$$\hat{\mu}_{\hat{\alpha}} = \frac{1}{B}\sum_{i=1}^{B} \hat{\alpha}^{*B}$$

Example

A thermostat used in an electrical devise is to be checked for the accuracy of its design setting of 200°F. Ten thermostats were tested to determine their actual settings, resulting in the following data:

202.2	203.4	200.5	202.5	206.3	198.0	203.7	200.8	201.3	199.0

- We wish to estimate the true mean of this thermostat.
- To understand the estimate we want to determine also the **standard error**.
- The standard error may be used to judge the precision of the statistic and/or calculate a confidence interval for the parameter that the statistic is estimating.

Estimate of $\mu = \hat{\mu} = \bar{X} = \frac{1}{10}\sum_{i=1}^{n} x_i = 201.77$

$\widehat{SE}(\bar{X}) = \frac{s}{\sqrt{n}} = \frac{2.4102}{\sqrt{10}} = 0.7621$

CI: $\bar{x} \pm t_{\alpha/2, n-1} SE(\bar{x})$

```
mean(temp) + c(-1,1)*qt(1.95/2,9)*sd(temp)/sqrt(10)
[1] 200.0459 203.4941
```

CI: $[200.05, 203.49]$

Assuming $X \sim N(\mu, \sigma)$

```
> shapiro.test(temp)
```
$\Rightarrow$ Test if X is Normal

```
Shapiro-Wilk normality test

data:  temp
W = 0.98453, p-value = 0.9848
```

H_0: Data is Normal H_A: Data is not Normal

Since p-value > 0.05, we fail to reject H_0.

Thus there is no evidence that the data is not Normal.

Booststrap for Estimating Standard Error

- Let $x_1, x_2, \ldots, x_n$ be a random sample from a probability distribution F with mean μ and standard deviation σ.
- Consider a very simplistic statistic, the sample mean $\bar{x}$. We know the **estimated standard error** of the mean is:

$$SE(\bar{x}) = \frac{s}{\sqrt{n}} = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n(n-1)}}$$

- So SE($\bar{x}$) can be readily calculated and there is not need to estimate.
- However, there are no such simple formulas for more complicated sample statistics, as in trimmed mean or sample median.
- To explain more we will try to estimate SE($\bar{x}$).
- For our example: $\bar{x} = 201.77$ and $SE(\bar{x}) = 0.762168$.

Example of Resampling in R

```
#Resample
temp = c(202.2,203.4,200.5,202.5,206.3,198.0,203.7,200.8,201.3,199.0)
B = 1000 #number of resamples
M = NA #a vector of the means
for(i in 1:B) {
    x = sample(temp,length(temp),replace=T)
    M[i] = mean(x)
    }
x #last sample in the for loop
```

```
 [1] 202.5 200.8 206.3 203.7 200.5 200.8 202.2 203.7 206.3 203.7
```

```
mean(x) #mean of the last sample
```

```
[1] 203.05
```

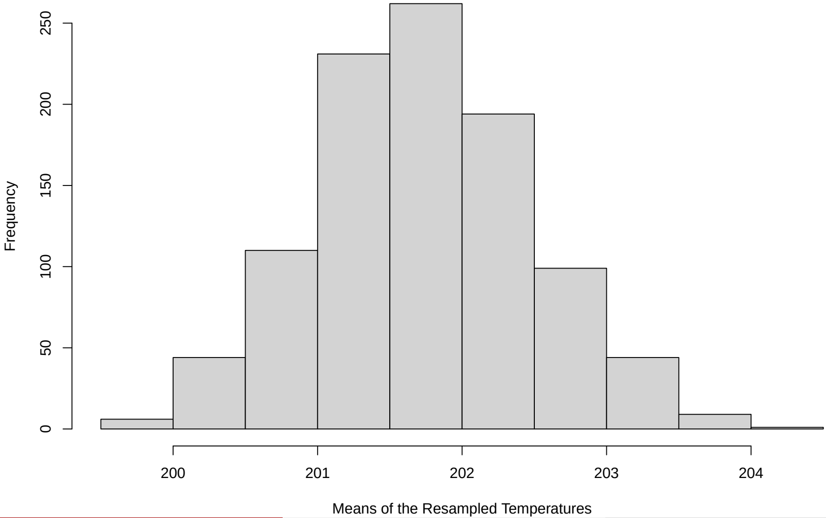
```
mean(M) #mean of the 1000 resampled means
```

```
[1] 201.7322
```

```
sd(M) #The estimated standard error of the mean
```

```
[1] 0.7490639
```

Histogram of the Means



The `boot` Function in `R`

Performing a bootstrap analysis in `R` entails only two steps:

1. Create a function that computes that statistic of interest.
2. Use the `boot()` function, which is part of the `boot` library, to perform the bootstrap by repeatedly sampling observations form the data set with replacement.

Example of Thermostat Themperature

```
#Bootstrap function
library(boot) #Uses the boot library
mean.fun <- function(dat, idx) mean(dat[idx], na.rm = TRUE)
boot.out = boot(data = temp,statistic = mean.fun,R = 1000)
boot.out
```

```
ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = temp, statistic = mean.fun, R = 1000)


Bootstrap Statistics :
    original  bias    std. error
t1*   201.77 0.00943   0.7119147
```

```
Bootstrap Statistics :
    original  bias    std. error
t1*   201.77 0.02078   0.7143009
```

```
mean(boot.out$t)  #mean of the means
```

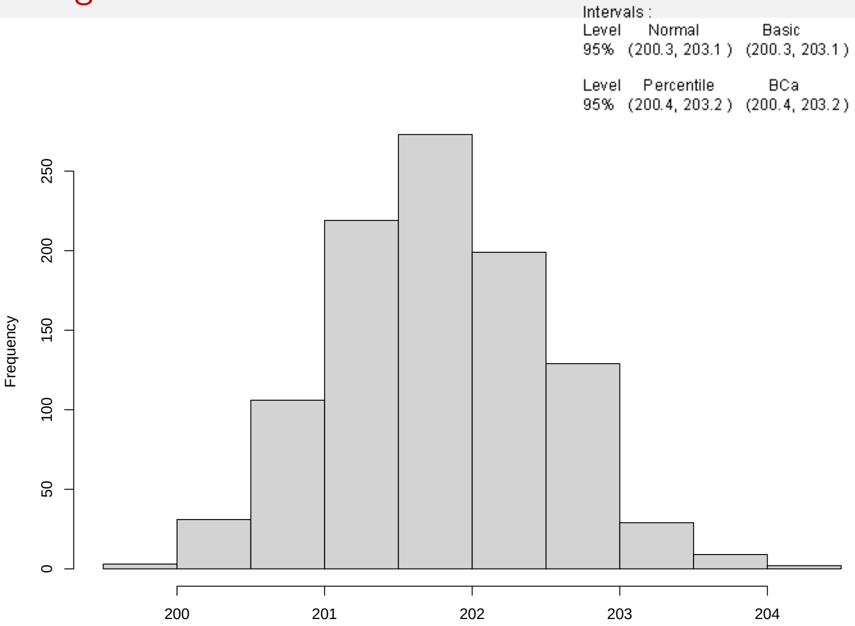
```
[1] 201.7794
```

- 201.77 = 0.0094

```
sd(boot.out$t)  #estimated standard error of the means
```

```
[1] 0.7119147
```

Histogram


Intervals :
Level Normal Basic
95% (200.3, 203.1) (200.3, 203.1)
Level Percentile BCa
95% (200.4, 203.2) (200.4, 203.2)
Frequency
0
50
100
150
200
250
200
201
202
203
204

The Ideal and Reality in Statistics World

Ideal World

- A standard error of our sample mean can be easily estimated and can find the estimated standard error.
- We assume we know or can estimate about the estimator's population.

Real World

- Hard to know the information about the population or it's distribution.
- The standard error of an estimate is hard to evaluate in general.

When the assumptions are violated, or when no formula exists for estimating standard errors, bootstrap is the powerful choice.

Why Does the Simulation of the Bootstrap Work?

Let $X_1, X_2, \ldots, x_n$ be a random sample from a population P with cumulative distribution function F. And let $M = g(X_1, X_2, \ldots, X_n)$ be our statistic for the parameter of interest. What we desire to is to know $Var(M)$. We resample B times.

By the **Law of Large Numbers**:

$$\bar{m} = \frac{1}{B}\sum_{j=1}^{B} M_j \xrightarrow{P} E(M), \text{ as } B \to \infty$$

Where $E(M)$ is the true mean of the statistic M.

In addition, the sample variance of these B statistics converges to the true variance of statistic M as $B \to \infty$.

$$s^2 = \frac{\sum_{j=1}^{B}(M_j - \bar{m})^2}{B-1} \xrightarrow{P} Var(M), \text{ as } B \to \infty$$

Where $Var(M)$ is the true variance of the statistic M.

Determine the Median Temperature

```
median.fun = function(dat, idx) median(dat[idx],na.rm = TRUE)
boot.out.median = boot(data = temp, statistic = median.fun, R = 1000)
boot.out.median
```

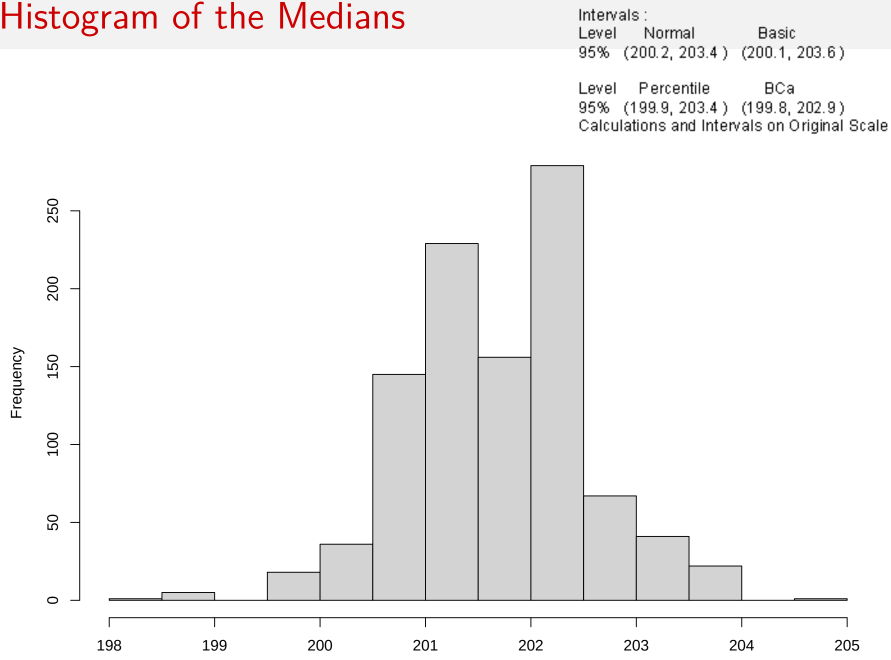
```
ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = temp, statistic = median.fun, R = 1000)


Bootstrap Statistics :
    original  bias    std. error
t1*   201.75  0.0081   0.8690654
```

From the original data: median = 201.75

From 1000 bootstraps: Mean(1000 medians) = 201.75 + 0.0081
= 201.7581

From 1000 bootstraps: SD(1000 medians) = 0.869

Histogram of the Medians



Example 2

The bootstrap approach can be used to assess the variability of the coefficient estimates and predictions from a statistical learning method. We we use the bootstrap approach in order to assess the variability of the estimates for β_0 and β_1, the intercept and slope terms for the linear regression model that uses horsepower to predict mpg in the Auto data set from the ISLR library.

Step 1: Create a function

- Create a function called boot.fn()
- This takes in the Auto data set as well as a set of indices for the observations, and returns the intercept and slope estimates for the linear regression model.

```
library(ISLR)
boot.fn = function(data,index)
  return(coef(lm(mpg~horsepower,data = data,subset = index)))
boot.fn(Auto,1:392)
```

```
(Intercept)  horsepower
 39.9358610  -0.1578447
```

$\hat{mpg} = 39.93586 - 0.15784\, hp$

Note: We do not need { and } at the beginning and end of the defined function because it is only one line.

Run the following command twice give two of the bootstrap estimates for the intercept and the slope.

```r
boot.fn(Auto,sample(392,392,replace = TRUE))
```

Do we get the same values when we run this function?

```
> boot.fn(Auto,sample(392,392,replace = TRUE))
(Intercept)  horsepower
 40.4510464 -0.1642299
> boot.fn(Auto,sample(392,392,replace = TRUE))
(Intercept)  horsepower
 40.3367926 -0.1599475
```

Step 2: Use the boot function

We can use the boot() function to compute the standard errors of 1000 bootstrap estimates for the intercept and slope terms.

```
boot.out = boot(Auto,boot.fn,1000)
boot.out
```

```
ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = Auto, statistic = boot.fn, R = 1000)


Bootstrap Statistics :
      original        bias    std. error
t1* 39.9358610  0.0254261104 0.855509708
t2* -0.1578447 -0.0003615604 0.007325578
```

y-intercept

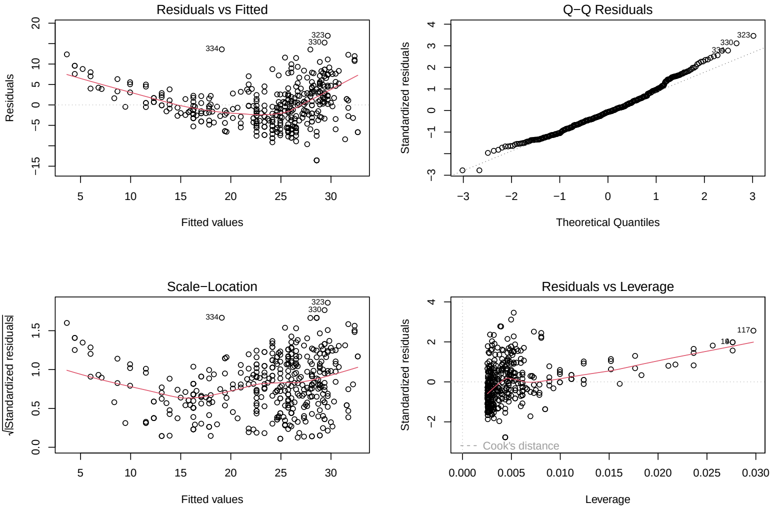
slope

The command below gives the estimates of the original data using the lm() function.

```
auto.lm = lm(mpg~horsepower,data = Auto)
summary(auto.lm)
```

```

Call:
lm(formula = mpg ~ horsepower, data = Auto)

Residuals:
     Min       1Q   Median       3Q      Max
-13.5710  -3.2592  -0.3435   2.7630  16.9240

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 39.935861   0.717499   55.66   <2e-16 ***
horsepower  -0.157845   0.006446  -24.49   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.906 on 390 degrees of freedom
Multiple R-squared:  0.6059,    Adjusted R-squared:  0.6049
F-statistic: 599.7 on 1 and 390 DF,  p-value: < 2.2e-16
```

Diagnostic Plots



Consider a Quadratic Model

We see that diagnostics plot shows that the assumptions are not met. We will see if a quadratic model is better.

$$mpg = \beta_0 + \beta_1 \times horsepower + \beta_2 \times horsepower^2 + \epsilon$$

```
boot.fn <- function(data , index)
  coef(lm(mpg ~ horsepower + I(horsepower ^2),
          data = data , subset = index))
boot(Auto , boot.fn, 1000)
summary(lm(mpg~ horsepower + I(horsepower^2), data = Auto))$coef
```

Results

```
ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = Auto, statistic = boot.fn, R = 1000)


Bootstrap Statistics :
       original        bias     std. error
t1* 56.900099702  5.912122e-02 2.0817702950
t2* -0.466189630 -1.310612e-03 0.0335199723
t3*  0.001230536  5.986143e-06 0.0001216492

                   Estimate   Std. Error   t value      Pr(>|t|)
(Intercept)     56.900099702 1.8004268063  31.60367 1.740911e-109
horsepower      -0.466189630 0.0311246171 -14.97816  2.289429e-40
I(horsepower^2)  0.001230536 0.0001220759  10.08009  2.196340e-21
```

Notice: These standard errors are closer.