The bootstrap

Jeffrey Leek, Assistant Professor of Biostatistics
Johns Hopkins Bloomberg School of Public Health

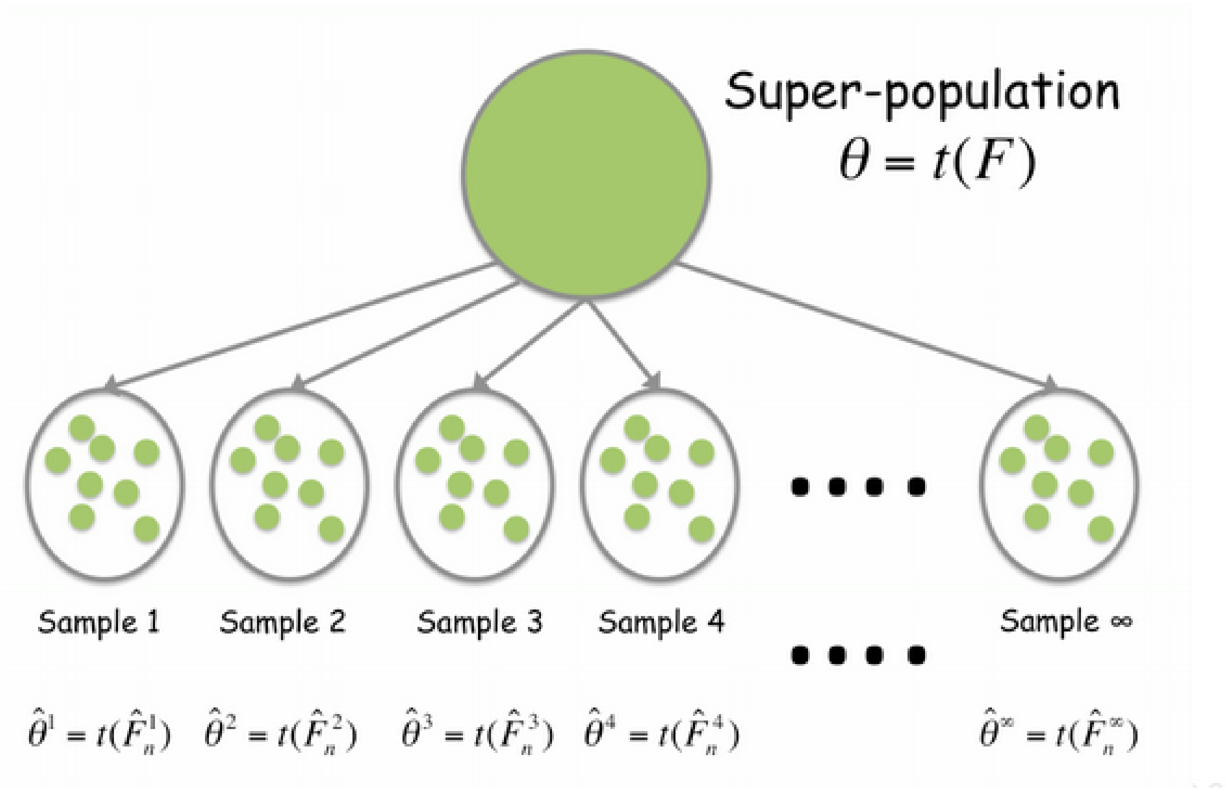
Key ideas

- Treat the sample as if it were the population

What it is good for:

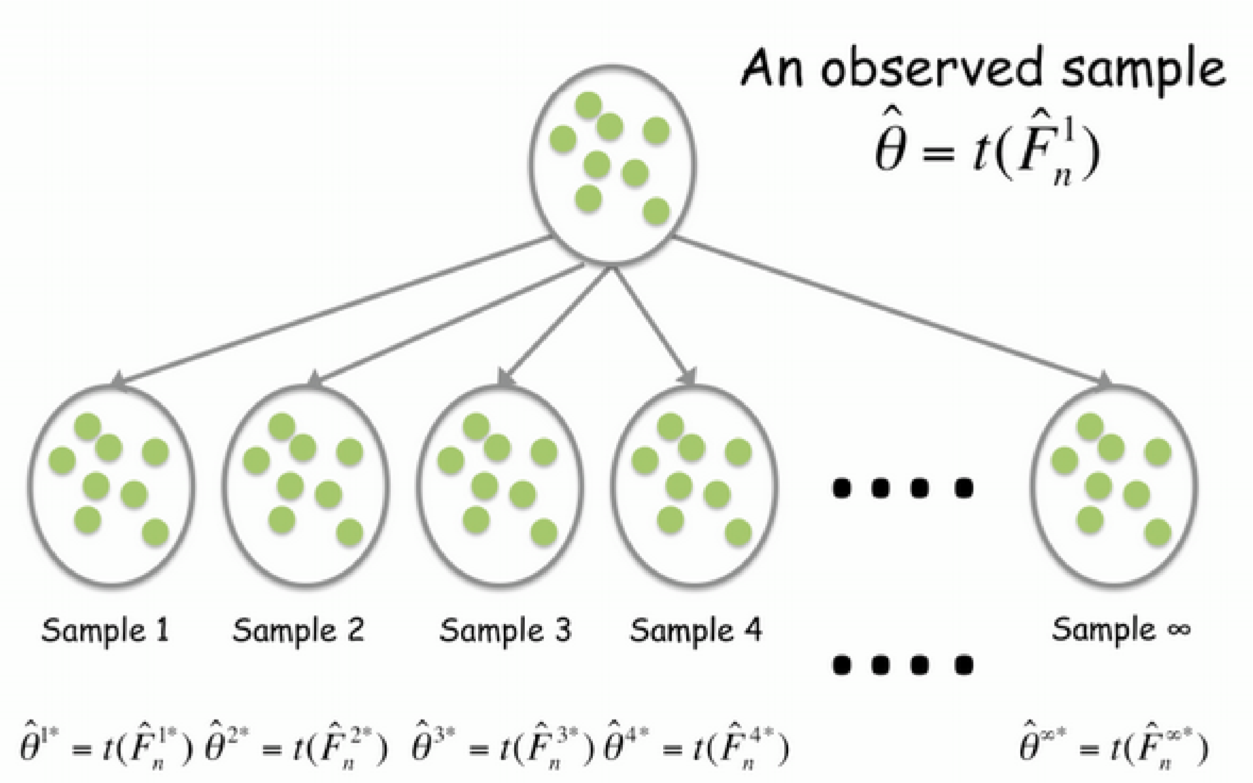
- Calculating standard errors
- Forming confidence intervals
- Performing hypothesis tests
- Improving predictors

The "Central Dogma" of statistics


Super-population
$\theta = t(F)$
Sample 1
Sample 2
Sample 3
Sample 4
Sample ∞
$\hat{\theta}^1 = t(\hat{F}_n^1)$
$\hat{\theta}^2 = t(\hat{F}_n^2)$
$\hat{\theta}^3 = t(\hat{F}_n^3)$
$\hat{\theta}^4 = t(\hat{F}_n^4)$
$\hat{\theta}^\infty = t(\hat{F}_n^\infty)$

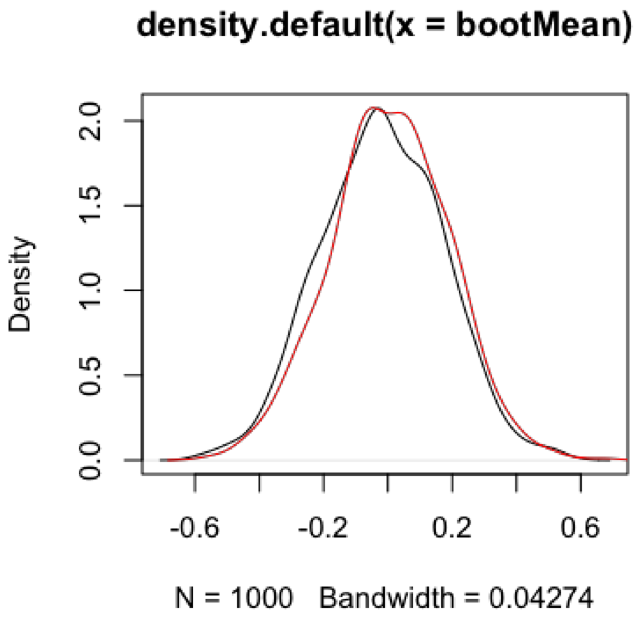
http://www.gs.washington.edu/academics/courses/akey/56008/lecture/lecture5.pdf

The bootstrap


An observed sample
$\hat{\theta} = t(\hat{F}_n^1)$
Sample 1
Sample 2
Sample 3
Sample 4
Sample ∞
$\hat{\theta}^{1*} = t(\hat{F}_n^{1*})$
$\hat{\theta}^{2*} = t(\hat{F}_n^{2*})$
$\hat{\theta}^{3*} = t(\hat{F}_n^{3*})$
$\hat{\theta}^{4*} = t(\hat{F}_n^{4*})$
$\hat{\theta}^{\infty*} = t(\hat{F}_n^{\infty*})$

Example

```
set.seed(333); x <- rnorm(30)
bootMean <- rep(NA,1000); sampledMean <- rep(NA,1000)
for(i in 1:1000){bootMean[i] <- mean(sample(x,replace=TRUE))}
for(i in 1:1000){sampledMean[i] <- mean(rnorm(30))}
plot(density(bootMean)); lines(density(sampledMean),col="red")
```



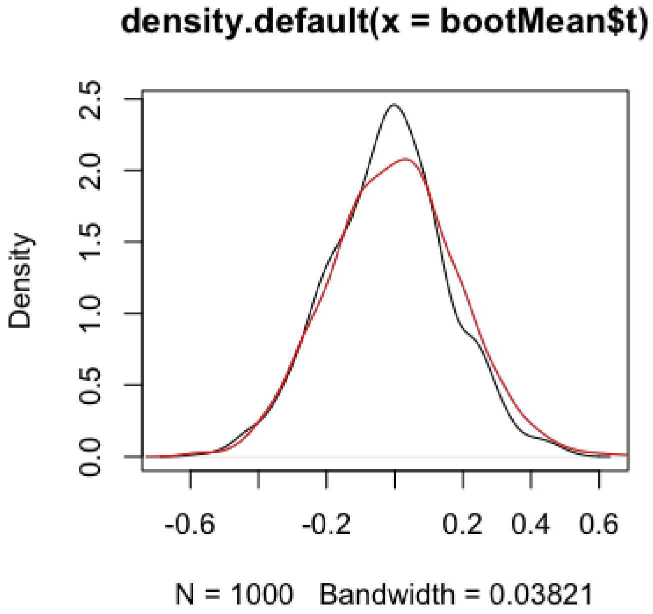
Example with boot package

```
set.seed(333); x <- rnorm(30); sampledMean <- rep(NA,1000)
for(i in 1:1000){sampledMean[i] <- mean(rnorm(30))}
meanFunc <- function(x,i){mean(x[i])}
bootMean <- boot(x,meanFunc,1000)
bootMean
```

```

ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = x, statistic = meanFunc, R = 1000)


Bootstrap Statistics :
    original     bias    std. error
t1* -0.01942 0.0006377       0.175
```

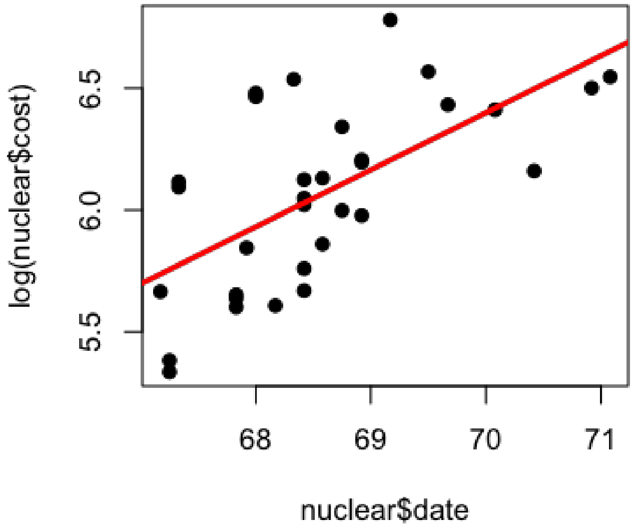
Plotting boot package example

```
plot(density(bootMean$t)); lines(density(sampledMean),col="red")
```



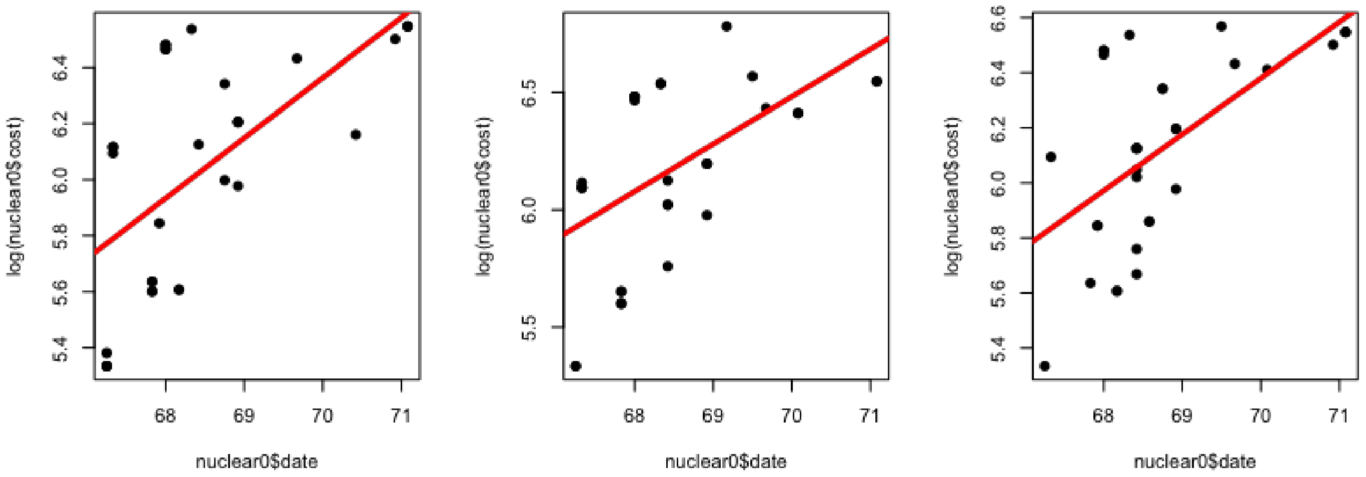
Nuclear costs

```
library(boot); data(nuclear)
nuke.lm <- lm(log(cost) ~ date,data=nuclear)
plot(nuclear$date,log(nuclear$cost),pch=19)
abline(nuke.lm,col="red",lwd=3)
```



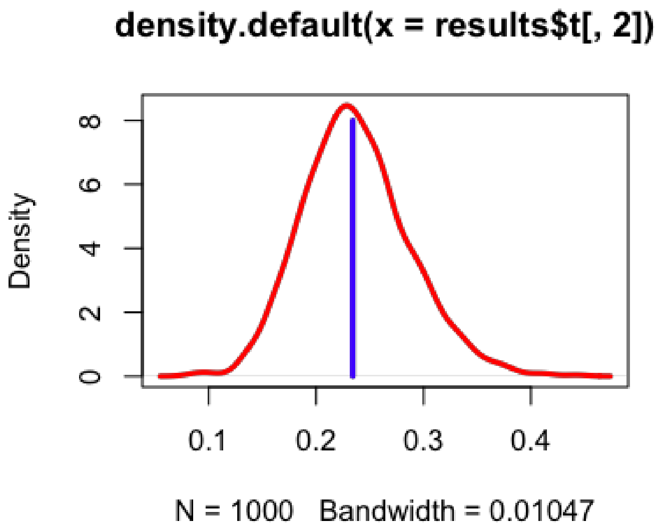
Nuclear costs

```
par(mfrow=c(1,3))
for(i in 1:3){
  nuclear0 <- nuclear[sample(1:dim(nuclear)[1],replace=TRUE),]
  nuke.lm0 <- lm(log(cost) ~ date,data=nuclear0)
  plot(nuclear0$date,log(nuclear0$cost),pch=19)
  abline(nuke.lm0,col="red",lwd=3)
}
```



Bootstrap distribution

```
bs <- function(data, indices,formula) {
  d <- data[indices,];fit <- lm(formula, data=d);return(coef(fit))
}
results <- boot(data=nuclear, statistic=bs, R=1000, formula=log(cost) ~ date)
plot(density(results$t[,2]),col="red",lwd=3)
lines(rep(nuke.lm$coeff[2],10),seq(0,8,length=10),col="blue",lwd=3)
```



http://www.statmethods.net/advstats/bootstrapping.html

Bootstrap confidence intervals

```
boot.ci(results)
```

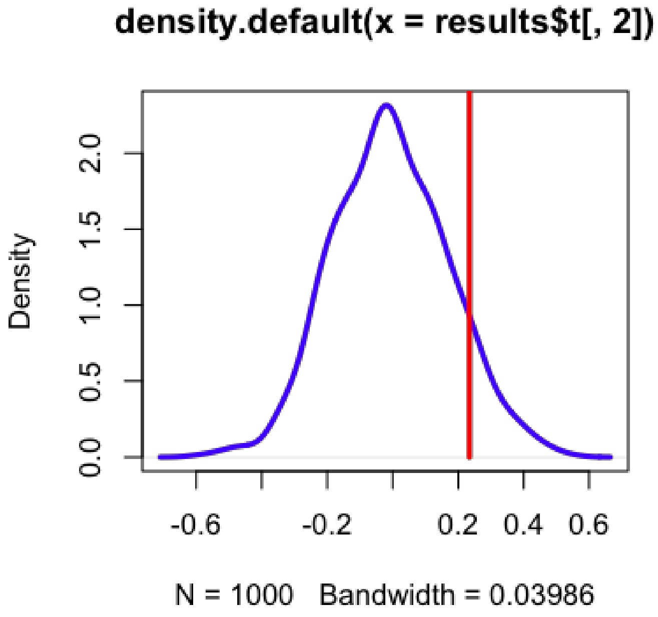
```
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 1000 bootstrap replicates

CALL :
boot.ci(boot.out = results)

Intervals :
Level      Normal              Basic             Studentized
95%   (-16.481,  -3.130 )   (-15.746,  -2.553 )   (-17.153,  -3.842 )

Level     Percentile            BCa
95%   (-17.435,  -4.242 )   (-17.475,  -4.249 )
Calculations and Intervals on Original Scale
```

Bootstrapping from a model

```
resid <- rstudent(nuke.lm)
fit0 <- fitted(lm(log(cost) ~ 1,data=nuclear))
newNuc <- cbind(nuclear,resid=resid,fit0=fit0)
bs <- function(data, indices) {
  return(coef(glm(data$fit0 + data$resid[indices] ~ data$date,data=data)))
}
results <- boot(data=newNuc, statistic=bs, R=1000)
```

Results

```
plot(density(results$t[,2]),lwd=3,col="blue")
lines(rep(coef(nuke.lm)[2],10),seq(0,3,length=10),col="red",lwd=3)
```



An empirical p-value

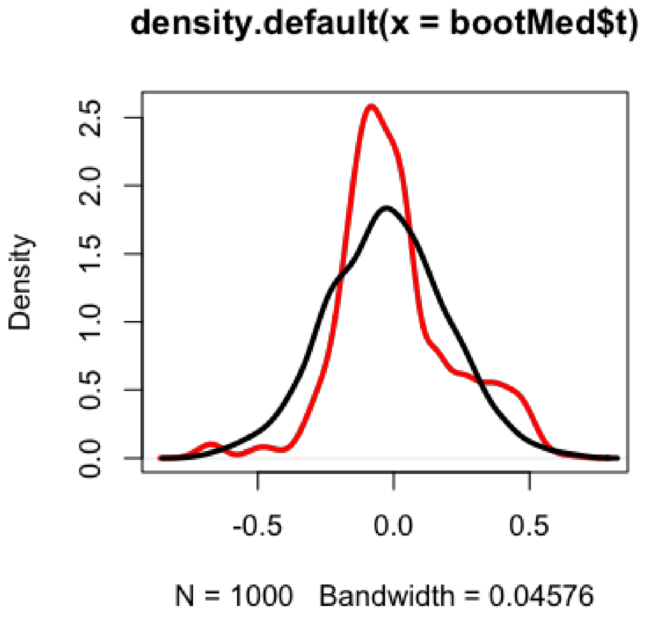
$$\hat{p} = \frac{1 + \sum_{b=1}^{B} |t_b^0| > |t|}{B + 1}$$

```
B <- dim(results$t)[1]
(1 + sum((abs(results$t[,2]) > abs(coef(nuke.lm)[2]))))/(B+1)
```

```
[1] 0.1838
```

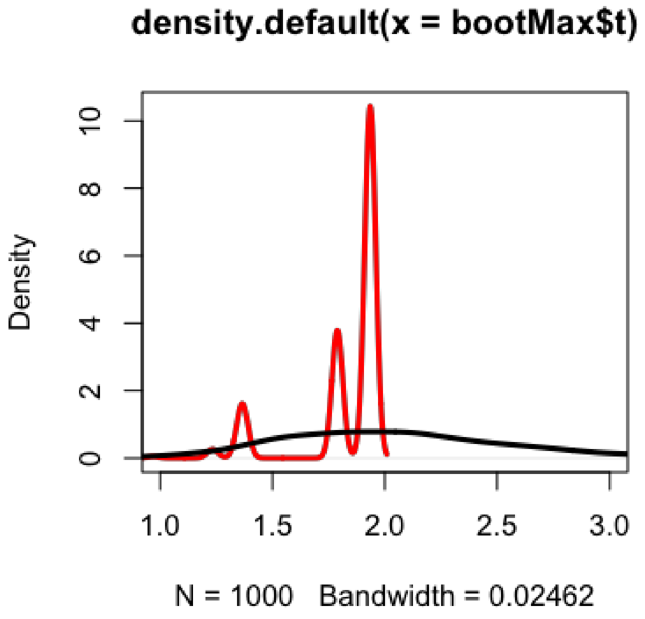
Bootstrapping non-linear statistics

```r
set.seed(555); x <- rnorm(30); sampledMed <- rep(NA,1000)
for(i in 1:1000){sampledMed[i] <- median(rnorm(30))}
medFunc <- function(x,i){median(x[i])}; bootMed <- boot(x,medFunc,1000)
plot(density(bootMed$t),col="red",lwd=3)
lines(density(sampledMed),lwd=3)
```



Things you can't bootstrap (max)

```
set.seed(333); x <- rnorm(30); sampledMax <- rep(NA,1000)
for(i in 1:1000){sampledMax[i] <- max(rnorm(30))}
maxFunc <- function(x,i){max(x[i])}; bootMax <- boot(x,maxFunc,1000)
plot(density(bootMax$t),col="red",lwd=3,xlim=c(1,3))
lines(density(sampledMax),lwd=3)
```



Notes and further resources

Notes:

- Can be useful for complicated statistics
- Be careful near the boundaries
- Be careful with non-linear functions

Further resources:

- Brian Caffo's bootstrap notes
- Nice basic intro to boot package
- Another basic boot tutorial
- An introduction to the bootstrap
- Confidence limits on phylogenies