Bootstrap

Canhong Wen

Agenda

- Toy collector solution
- Plug-In and the Bootstrap
- Nonparametric and Parametric Bootstraps
- Examples

Example: Toy Collector

Children (and some adults) are frequently enticed to buy breakfast cereal in an effort to collect all the action figures. Assume there are 15 action figures and each cereal box contains exactly one with each figure being equally likely.

- Find the expected number of boxes needed to collect all 15 action figures.
- Find the standard deviation of the number of boxes needed to collect all 15 action figures.

- Consider the probability of the "new toy" given we already have i toys
- Then
$$P(\text{New Toy}|i) = \frac{15-i}{15}$$
- Then since each box is independent, our waiting time until a "new toy" is a geometric random variable
- The mean is
$$\frac{15}{15} + \frac{15}{14} + \cdots + \frac{15}{1} \approx 49.77$$
- The variance is
$$\frac{15(1-15/15)}{15} + \frac{15(1-14/15)}{14} + \cdots + \frac{15(1-1/15)}{1} \approx 34.77$$
with standard deviation 5.90

Example: Toy Collector

- Now suppose we no longer have equal probabilities, instead let

Figure	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Probability	.2	.1	.1	.1	.1	.1	.05	.05	.05	.05	.02	.02	.02	.02	.02

- Estimate the expected number of boxes needed to collect all 15 action figures.
- What is the uncertainty of your estimate?
- What is the probability you bought more than 300 boxes? 500 boxes? 800 boxes?

Example: Toy Collector

```
prob.table <- c(.2, .1, .1, .1, .1, .1, .05, .05, .05, .05, .02, .02, .02, .02, .02)
prob.table <- rep(1, 15)/15
boxes <- seq(1,15)
box.count <- function(prob=prob.table){
  check <- double(length(prob))
  i <- 0
  while(sum(check)<length(prob)){
    x <- sample(boxes, 1, prob=prob)
    check[x] <- 1
    i <- i+1
  }
  return(i)
}
```

Example: Toy Collector

```
trials <- 1000
sim.boxes <- double(trials)
for(i in 1:trials){
  sim.boxes[i] <- box.count()
}
est <- mean(sim.boxes)
sd(sim.boxes)
```

```
## [1] 16.65079
```

```
mcse <- sd(sim.boxes) / sqrt(trials)
interval <- est + c(-1,1)*1.96*mcse
est
```

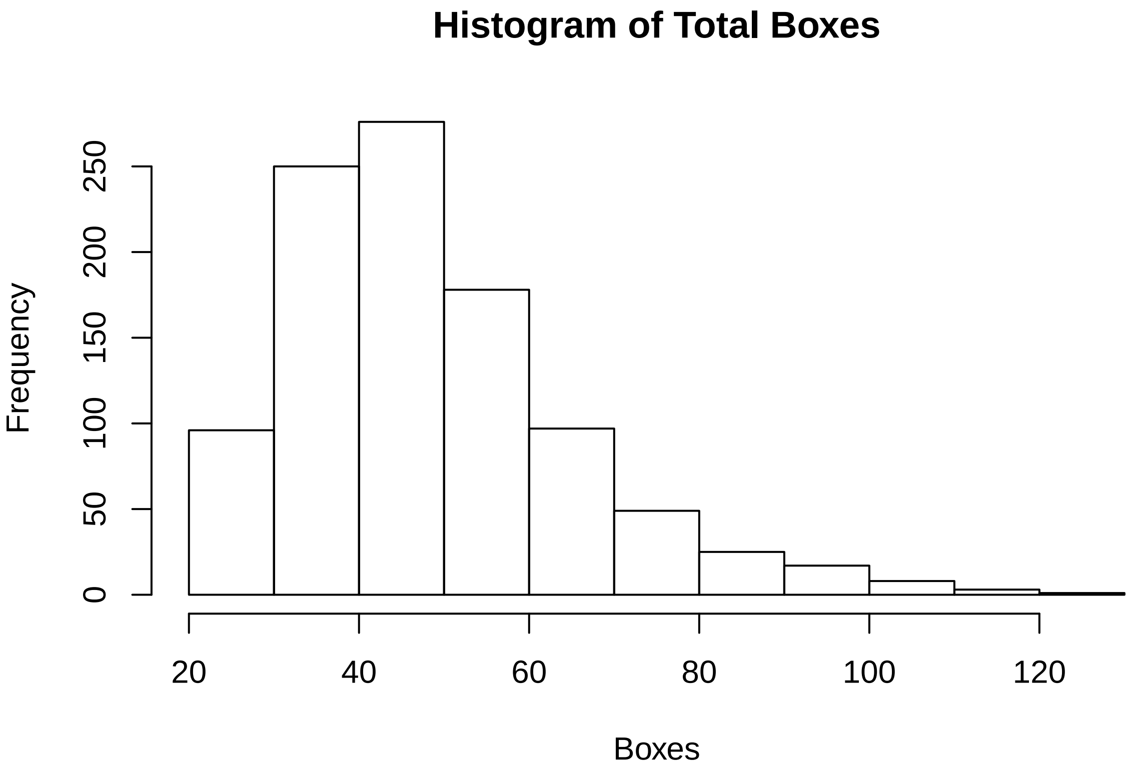
```
## [1] 49.107
```

```
interval
```

```
## [1] 48.07497 50.13903
```

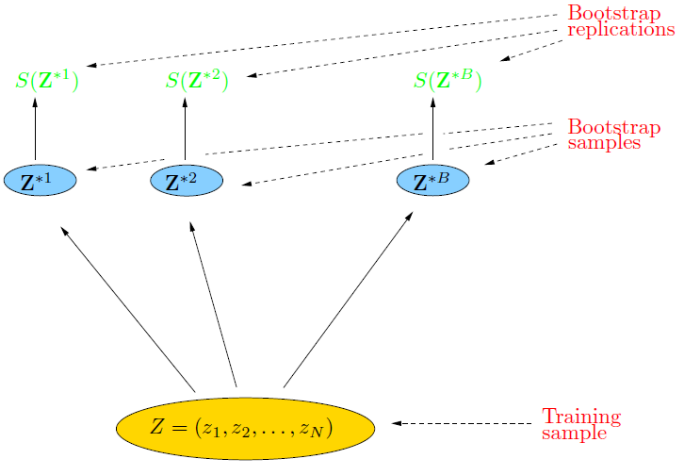
Example: Toy Collector

```
hist(sim.boxes, main="Histogram of Total Boxes", xlab="Boxes")
abline(v=300, col="red", lwd=2)
```



Bootstrap

- Bootstrapping is a computational intensive method that allows researchers to simulate the distribution of a statistic.
- The idea is to repeatedly resample the observed data, each time producing an empirical distribution function from the resampled data.
- For each resampled data set or equivalently each empirical distribution function, a new value of the statistic can be computed, and the collection of these values provides an estimate of the sampling distribution of the statistic of interest.



Nonparametric Bootstrap

World	Real	Bootstrap
true distribution	F	$\hat{F}_n$
data	$X_1, \ldots, X_n$ i.i.d. F	$X_1^*, \ldots, X_n^*$ i.i.d. $\hat{F}_n$
empirical distribution	$\hat{F}_n$	F_n^*
parameter	$\theta = t(F)$	$\hat{\theta}_n = t(\hat{F}_n)$
estimator	$\hat{\theta}_n = t(\hat{F}_n)$	$\theta_n^* = t(F_n^*)$
error	$\hat{\theta}_n - \theta$	$\theta_n^* - \hat{\theta}_n$
standardized error	$\frac{\hat{\theta}_n - \theta}{s(\hat{F}_n)}$	$\frac{\theta_n^* - \hat{\theta}_n}{s(F_n^*)}$

Notation $\theta = t(F)$ means θ is some function of the true unknown distribution

Nonparametric Bootstrap

- The notation $X_1^*, \ldots, X_n^*$ i.i.d. $\hat{F}_n$ means $X_1^*, \ldots, X_n^*$ are independent and identically distributed from the empirical distribution of the real data
- Sampling from the empirical distribution is just like sampling from a finite population, where the population is the real data $X_1, \ldots, X_n$
  - To be i.i.d. sampling must be with replacement
  - $X_1^*, \ldots, X_n^*$ are a sample with replacement from $X_1, \ldots, X_n$
  - Called resampling

Nonparametric Bootstrap

- We want to know the sampling distribution of $\hat{\theta}_n$ or $\hat{\theta}_n - \theta$ or $\frac{\hat{\theta}_n - \theta}{s(\hat{F}_n)}$
  - This sampling distribution depends on the true unknown distribution F of the real data
  - May be very difficult or impossible to calculate theoretically
  - Even asymptotic approximation may be difficult, if the parameter $\theta = t(F)$ is a sufficiently complicated function of the true unknown F
  - The statistical theory we have covered is quite amazing in what it does, but there is a lot it doesn't do

Bootstrap Estimate of Standard Error

A bootstrap estimate of the standard error is

$$\hat{se}(\hat{\theta}^*) = \sqrt{\frac{1}{B-1} \sum_{b=1}^{B} (\hat{\theta}^{(b)} - \bar{\hat{\theta}}^*)^2},$$

where $\bar{\hat{\theta}}^* = \frac{1}{B} \sum_{b=1}^{B} \hat{\theta}^{(b)}$.

```r
library(bootstrap)
print(cor(law$LSAT, law$GPA))      # Small data set
```

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

```r
print(cor(law82$LSAT, law82$GPA))  # Full data
```

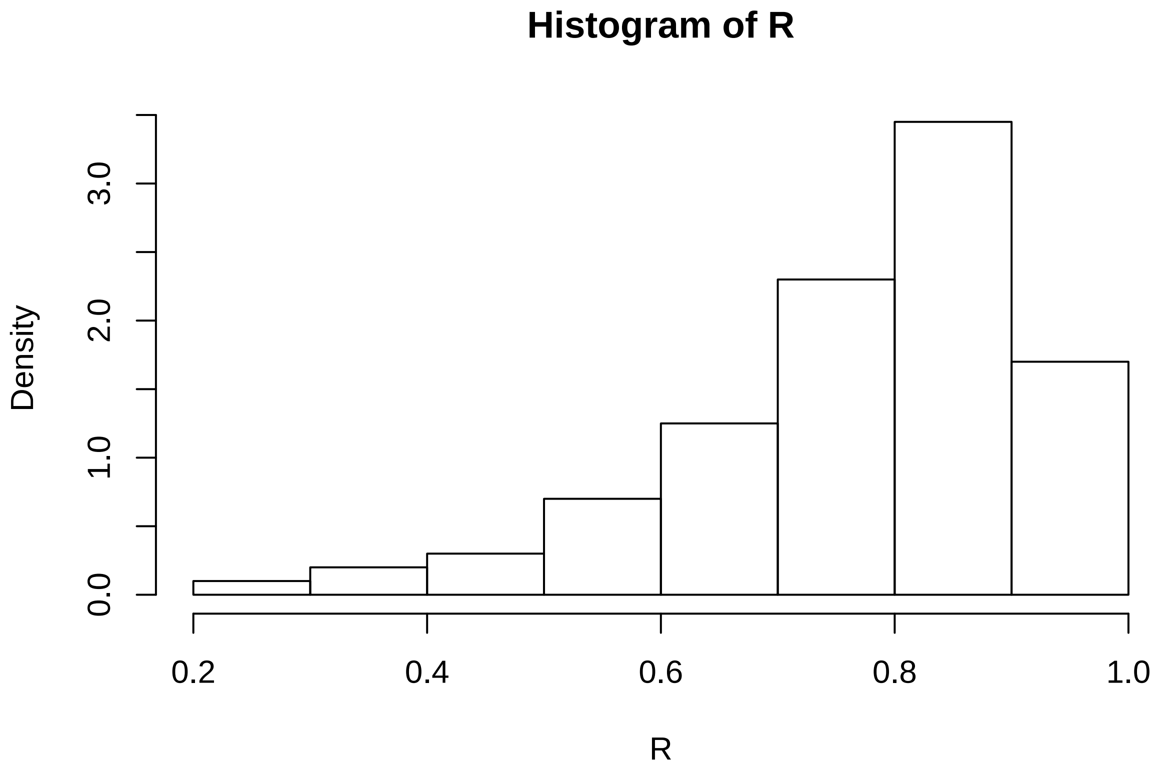
```
## [1] 0.7599979
```

```r
# Set up the bootstrap
B <- 200             # Number of bootstrapping
n <- nrow(law)       # Sample size
R <- numeric(B)

# Bootstrap estimate of standard error of Pearson correlation
for(b in 1:B){
  i <- sample(1:n, size=n, replace = TRUE) # i is a vector of indices
  LSAT <- law$LSAT[i]
  GPA <- law$GPA[i]
  R[b] <- cor(LSAT, GPA)
}
print(se.R <- sd(R))
```

```
## [1] 0.146413
```

```r
hist(R, probability = TRUE)
```



We can also use the built in bootstrapping functions, see `boot` and `bootstrap` library.

```
library(boot)
r <- function(x, i) cor(x[i,1], x[i,2])
obj <- boot(data = law, statistic = r, R = 200)
obj
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = law, statistic = r, R = 200)
##
##
## Bootstrap Statistics :
##      original       bias    std. error
## t1* 0.7763745 -0.01205151    0.150183
```

```
sd(obj$t)
```

```
## [1] 0.150183
```

```
library(bootstrap)
theta <- function(x,xdata){ cor(xdata[x,1],xdata[x,2]) }
obj2 <- bootstrap(1:n, nboot = 200, theta = theta, law)
sd(obj2$thetastar)
```

```
## [1] 0.1298428
```

Bootstrap Estimate of bias

A bootstrap estimate of the bias is

$$bias(\hat{\theta}) = \bar{\hat{\theta}}^* - \hat{\theta}.$$

```
R.hat <- cor(law$LSAT, law$GPA)
bias <- mean(R - R.hat)
bias
```

```
## [1] -0.005619555
```

Bootstrap Confidence Intervals

- The first order normal approximation interval
- The bootstrap percentile interval
- The basic bootstrap interval
- The studentized bootstrap interval
- The accelerated bias-corrected percentile (BCa) interval.

First Order Normal Approximation Interval

- Assume $\hat{\theta}$ is an estimate of θ, and $E(\hat{\theta}) = \theta$.
- By CLT,

$$Z = \frac{\hat{\theta} - \theta}{se(\hat{\theta})} \sim \mathcal{N}(0,1), \text{ as } n \to \infty.$$

- Then the 100(1 - α)% approximation confidence interval is given by

$$(\hat{\theta} - z_{\alpha/2} se(\hat{\theta}),\ \hat{\theta} + z_{\alpha/2} se(\hat{\theta})),$$

  where $z_{\alpha/2} = \Phi^{-1}(1 - \alpha/2)$.

Bootstrap Percentile Interval

- Simplest method of making confidence intervals for the unknown parameter is to take $\alpha/2$ and $1 - \alpha/2$ quantiles of the bootstrap distribution of the estimator θ_n^* as endpoints of the 100(1-α)% confidence interval.
- That is, $100(1 - \alpha)\%$ bootstrap percentile interval is $(\hat{\theta}^*_{1-\alpha/2},\ \hat{\theta}^*_{\alpha/2})$.

Basic Bootstrap Interval

- $$P(L < \hat{\theta} - \theta < U) = 1 - \alpha$$
- Since the distribution of $\hat{\theta} - \theta$ is unknown, so we use bootstrap to estimate it. Then we have
$$P(\hat{\theta}^*_{1-\alpha/2} - \hat{\theta} < \hat{\theta} - \theta < \hat{\theta}^*_{\alpha/2} - \hat{\theta}) \approx 1 - \alpha$$
- Thus the 100(1 - α)% basic bootstrap confidence interval is given by
$$(2\hat{\theta} - \hat{\theta}_{\alpha/2},\ 2\hat{\theta} - \hat{\theta}_{1-\alpha/2})$$

Studentized Bootstrap Interval

- The 100(1 -α)% studentized bootstrap confidence interval is given by
$$(\hat{\theta} - t^*_{1-\alpha/2}\hat{se}(\hat{\theta}),\ \hat{\theta} + t^*_{\alpha/2}\hat{se}(\hat{\theta})),$$
where $\hat{se}(\hat{\theta})$, $t^*_{1-\alpha/2}$ and $t^*_{\alpha/2}$ are computed by Bootstrap.

Accelerated Bias-Corrected Percentile (BCa) Interval

- The 100(1 - α)% BCa confidence interval is $(\hat{\theta}^*_{\alpha_1},\ \hat{\theta}^*_{\alpha_2})$, where
$$\alpha_1 = \Phi\left(\hat{z}_0 + \frac{\hat{z}_0 + z_{1-\alpha/2}}{1 - \hat{a}(\hat{z}_0 + z_{1-\alpha/2})}\right), \alpha_2 = \Phi\left(\hat{z}_0 + \frac{\hat{z}_0 + z_{\alpha/2}}{1 - \hat{a}(\hat{z}_0 + z_{\alpha/2})}\right),$$
- Bias corrected factor: $\hat{z}_0 = \Phi^{-1}\left(\frac{1}{B}\sum_{b=1}^{B} I(\hat{\theta}^{(b)} < \hat{\theta})\right)$
- Acceleration factor:
$$\hat{a} = \frac{\sum_{i=1}^{n}(\hat{\theta}_{(i)} - \hat{\hat{\theta}}_{(\cdot)})^3}{6(\sum_{i=1}^{n}(\hat{\theta}_{(i)} - \hat{\hat{\theta}}_{(\cdot)})^2)^{3/2}}$$

R Code Demo

```r
library(boot)
data(law, package = "bootstrap")
boot.obj <- boot(law, R = 2000,
         statistic = function(x, i){cor(x[i,1], x[i,2])})
print(boot.ci(boot.obj, type=c("norm","basic", "perc", "bca")))
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 2000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot.obj, type = c("norm", "basic", "perc",
##     "bca"))
##
## Intervals :
## Level      Normal              Basic
## 95%   ( 0.5212,  1.0524 )   ( 0.5898,  1.1235 )
```

```
##
## Level     Percentile            BCa
## 95%   ( 0.4293,  0.9629 )   ( 0.3314,  0.9428 )
## Calculations and Intervals on Original Scale
```

Exercise: Calculate the studentized bootstrap interval.

Parametric Bootstrap

- The parametric bootstrap is just like the nonparametric bootstrap except for one difference in the analogy
- We use a parametric model $F_{\hat{\theta}_n}$ rather than the empirical distribution $\hat{F}_n$ as the analog of the true unknown distribution in the bootstrap world

Parametric Bootstrap

World	Real	Bootstrap
parameter	θ	$\hat{\theta}_n$
true distribution	F_θ	$F_{\hat{\theta}_n}$
data	$X_1, \dots, X_n$ i.i.d. F_θ	$X_1^*, \dots, X_n^*$ i.i.d. $F_{\hat{\theta}_n}$
estimator	$\hat{\theta}_n = t(X_1, \dots, X_n)$	$\theta_n^* = t(X_1^*, \dots, X_n^*)$
error	$\hat{\theta}_n - \theta$	$\theta_n^* - \hat{\theta}_n$
standardized error	$\frac{\hat{\theta}_n - \theta}{s(X_1, \dots, X_n)}$	$\frac{\theta_n^* - \hat{\theta}_n}{s(X_1^*, \dots, X_n^*)}$

Parametric Bootstrap

- Simulation from the parametric model $F_{\hat{\theta}_n}$ not analogous to finite population sampling and does not resample the data like the nonparametric bootstrap does
- Instead we simulate the parametric model
- May be easy (when R has a function to provide such random simulations) or difficult

Example: Air-Conditioning

In this example our aim is to test the hypothesis that the true value of the index is 1 (i.e. that the data come from an exponential distribution) against the alternative that the data come from a gamma distribution with index not equal to 1.

```
air.fun <- function(data) {
     ybar <- mean(data$hours)
     para <- c(log(ybar), mean(log(data$hours)))
     ll <- function(k) {
          if (k <= 0) 1e200 else lgamma(k)-k*(log(k)-1-para[1]+para[2])
     }
     khat <- nlm(ll, ybar^2/var(data$hours))$estimate
     c(ybar, khat)
}

air.rg <- function(data, mle) {
    # Function to generate random exponential variates.
    # mle will contain the mean of the original data
    out <- data
    out$hours <- rexp(nrow(out), 1/mle)
    out
}
```

```
air.boot <- boot(aircondit, air.fun, R = 999, sim = "parametric",
                 ran.gen = air.rg, mle = mean(aircondit$hours))

# The bootstrap p-value can then be approximated by
sum(abs(air.boot$t[,2]-1) > abs(air.boot$t0[2]-1))/(1+air.boot$R)
```

```
## [1] 0.438
```

Nonparametric versus Parametric

- The nonparametric bootstrap is nonparametric. That means it always does the right thing, except when it doesn't. It doesn't work when the sample size is too small or when the square root law doesn't hold or when the data are not IID or when various technical issues arise that are beyond the scope of this course – the parameter is not a nice enough function of the true unknown distribution.
- The parametric bootstrap is parametric. That means it is always wrong when the model is wrong. On the other hand, when the parametric bootstrap does the right thing (when the statistical model is correct), it does a much better job at smaller sample sizes than the nonparametric bootstrap.

Nonparametric versus Parametric

- When the parameter θ is defined in terms of the parametric statistical model and can only be estimated using the parametric model (by maximum likelihood perhaps), the statistical model is needs to be correct for the parameter estimate $\hat{\theta}_n$ to make sense
- Since we already need the statistical model to be correct, the parametric bootstrap is the logical choice

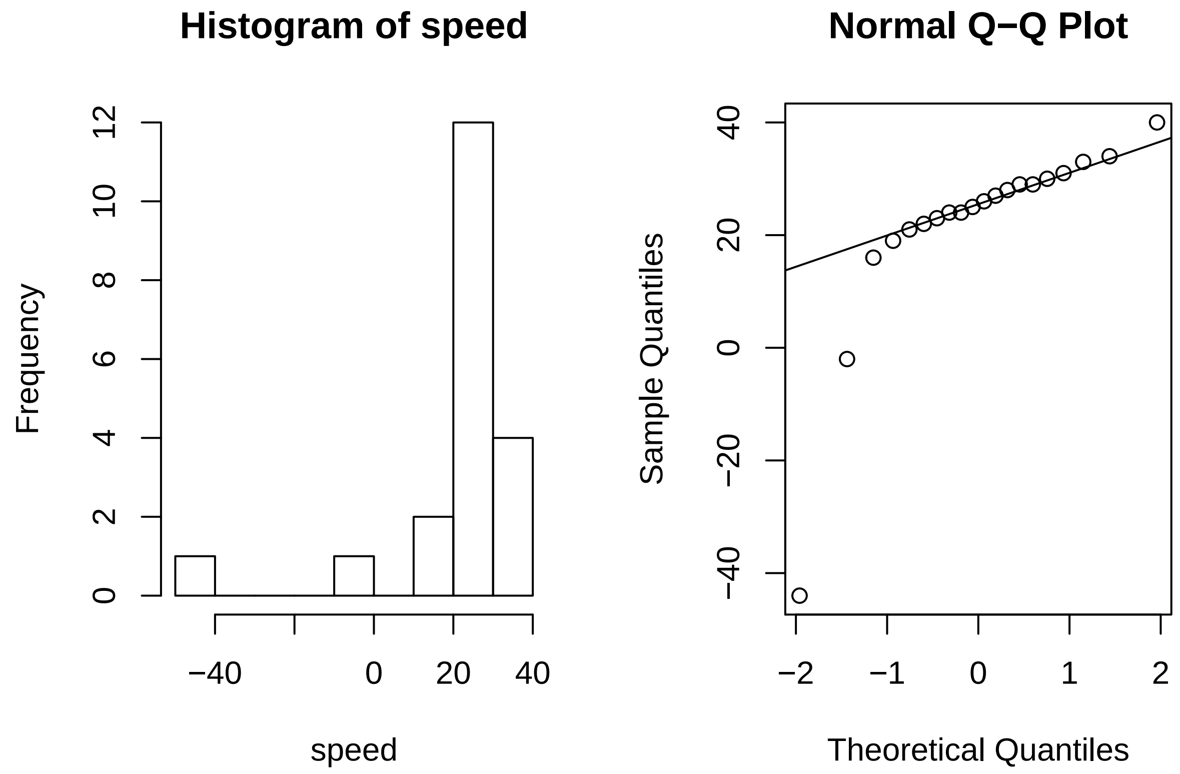
Abnormal speed of light data

- In 1882 Simon Newcomb performed an experiment to measure the speed of light
- Measured time it took for light to travel from Fort Myer on the west bank of the Potomac River to a fixed mirror at the foot of the Washington monument 3721 meters away
- In the units of the data, the currently accepted "true" speed of light is 33.02
- Does the data support the current accepted speed of 33.02?

```
speed <- c(28, -44, 29, 30, 26, 27, 22, 23, 33, 16, 24, 29, 24, 40 , 21, 31, 34, -2, 25, 19)
```

- To convert these units to time in the millionths of a second, multiply by 10^{-3} and add 24.8

Abnormal speed of light data



Abnormal speed of light data

- A *t*-test assumes the population of measurements is normally distributed
- With this small sample size and a severe departure from normality, we cann't be guaranteed a good approximation
- Instead, we can consider the bootstrap

Abnormal speed of light data

1. State null and alternative hypotheses

$$H_0 : \mu = 33.02 \text{ versus } H_a : \mu \neq 33.02$$

2. Choose a significance level, in our case 0.05
3. Choose a test statistic, since we wish to estimate the mean speed we can use the sample average
4. Find the observed value of the test statistic
5. Calculate a p-value?

```
mean(speed)
```

```
## [1] 21.75
```

Abnormal speed of light data

- We now need a p-value, but we don't have the sampling distribution of our test statistic when the null hypothesis is true
- It is approximately normal, but that is a poor approximation here
- Instead we can perform a simulation under conditions in which we know the null hypothesis is true
- Use our data to represent the population, but first we shift it over so that the mean really is 33.02

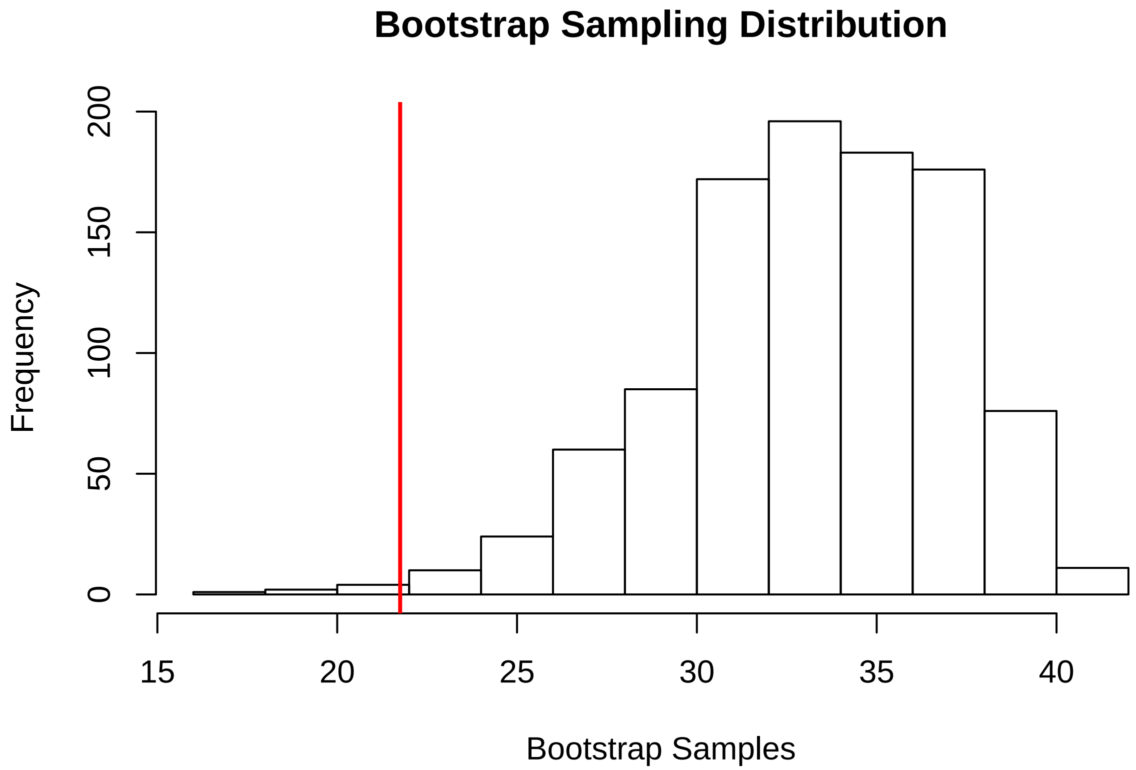
```
newspeed <- speed - mean(speed) + 33.02
```

- Histogram of `newspeed` will have exactly the same shape as speed, but will be shifted

Abnormal speed of light data

- Now we reach into our fake population and take out 20 observations at random, with replacement
- We take out 20 because that is the size of our initial sample
- We calculate the average and save it, then repeat this process many, many times
- Now we have a sampling distribution with mean 33.02
- Can compare this to our observed sample average and obtain a p-value

```
n <- 1000
bstrap <- double(n)
for (i in 1:n){
  newsample <- sample(newspeed, 20, replace=T)
  bstrap[i] <- mean(newsample)
  }
```



- Don't look normal, which means we did the right thing
- Not impossible for the sample average to be 21.75
- But it is not all that common, either

Abnormal speed of light data

- The p-value is the probability of getting something more extreme than what we observed
- Notice 21.75 is $33.02 - 21.75 = 11.27$ units away from the null hypothesis
- So p-value is the probability of being more than 11.27 units away from 33.02

```r
(sum(bstrap < 21.75) + sum(bstrap > 44.29))/1000
```

```
## [1] 0.006
```

- Since our significance level is 5%, we reject H_0 and conclude that Newcomb measurements were not consistent with the currently accepted figure

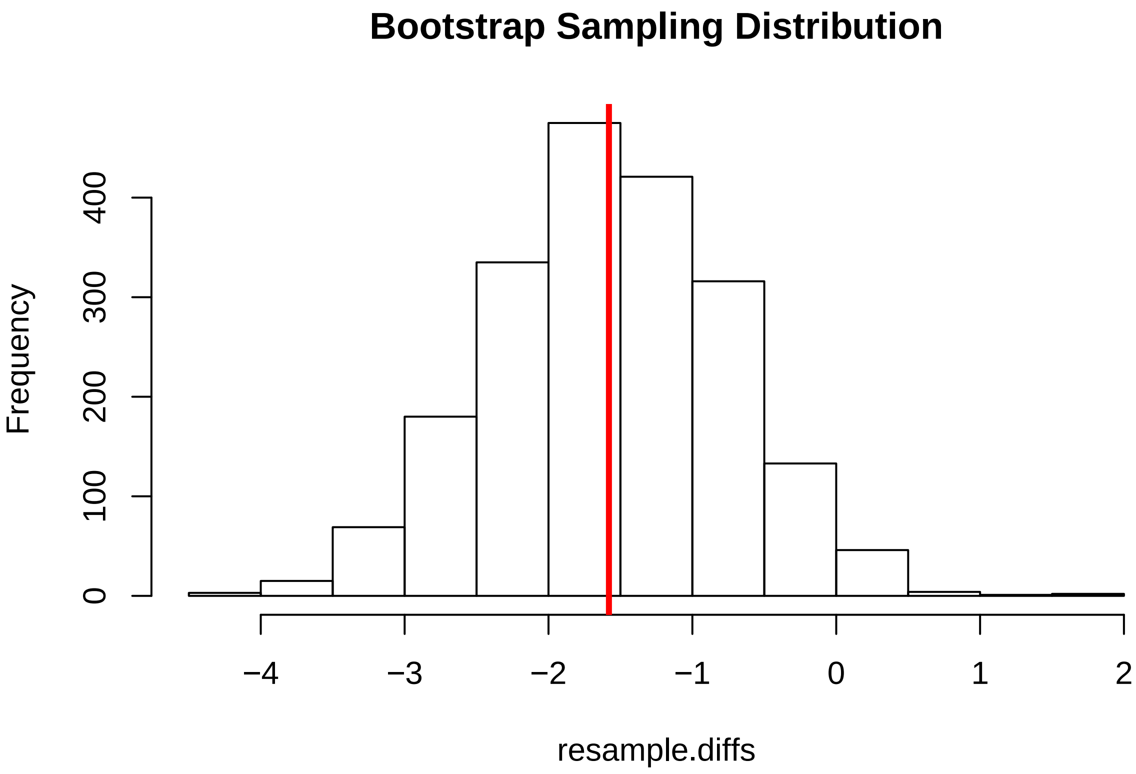
Example: Sleep study

- The two sample t-test checks for differences in means according to a known null distribution
- Similar to permutation tests
- Let's resample and generate the sampling distribution under the bootstrap assumption

```r
bootstrap.resample <- function (object) sample (object, length(object), replace=TRUE)
diff.in.means <- function(df) {
  mean(df[df$group==1,"extra"]) - mean(df[df$group==2,"extra"])
}
resample.diffs <- replicate(2000, diff.in.means(sleep[bootstrap.resample(1:nrow(sleep)),]))
```

Example: Sleep study

```r
hist(resample.diffs, main="Bootstrap Sampling Distribution")
abline(v=diff.in.means(sleep), col=2, lwd=3)
```



Summary

- Bootstrapping provides a nonparametric approach to statistical inference when distributional assumptions may not be met
- Enables calculation of standard errors and confidence intervals in a variety of situations, e.g. medians, correlation coefficients, regression parameters, . . .
- Hypothesis tests are a little more challenging
- The bootstrap is large sample, approximate, and asymptotic!
- Works when the empirical distribution $\hat{F}_n$ is close to the true unknown distribution F
- Usually the case when the sample size n is large and not otherwise, no method can save bad data!