An Introduction to the Bootstrap

For example, \( \theta = t(F) \) could be the

- mean,
- variance,
- standard deviation,
- coefficient of variation \( \left( \frac{SD}{|\text{Mean}|} \right) \),
- median or another quantile of a univariate distribution,
- correlation between components of a bivariate distribution, etc.

Suppose \( F \) is some population distribution function.

Suppose we are interested in some parameter of this distribution that can be written as

\[ \theta = t(F) \]

where \( \theta \) is a real value and \( t \) is a known function that maps distribution functions to \( \mathbb{R} \).

Statistics involves the estimation of population parameters like \( \theta = t(F) \) using sample statistics.

Suppose \( Y_1, Y_2, \ldots, Y_n \overset{iid}{\sim} F \).

Let \( \hat{\theta} = S(Y_1, \ldots, Y_n) \) be an estimator of \( \theta \) given by the sample statistic \( S(Y_1, \ldots, Y_n) \).
For example, if \( \Theta = \mu(F) \) is the mean of a univariate distribution defined by \( F \), we might choose

\[ \hat{\Theta} = \frac{1}{n} \sum_{i=1}^{n} Y_i = \bar{Y}. \]

as our estimator of \( \Theta \).

These questions are easy to answer in some cases using statistical theory.

For example, suppose we know that \( F \) is the distribution function of the \( N(\Theta, \sigma^2) \) distribution where \( \Theta \in \mathbb{R} \) is unknown. Then we know that

\[ \hat{\Theta} = \bar{Y} \sim N(\Theta, \frac{1}{n}). \]

How good is our estimator \( \hat{\Theta} \) of the population parameter \( \Theta \)?

Is it unbiased?

Does it have low variability?

Is it likely to be close to \( \Theta \)?

What is the distribution of \( \hat{\Theta} \)?

\[ \hat{\Theta} = \bar{Y} \sim N(\Theta, \frac{1}{n}) \implies \]

\[ E(\hat{\Theta}) = \Theta \quad (\hat{\Theta} \text{ is unbiased for } \Theta) \]

\[ \text{Var}(\hat{\Theta}) = \frac{\sigma^2}{n} \quad (\text{Variance of } \hat{\Theta} \text{ is low for large } n) \]

\[ \Pr(|\hat{\Theta} - \Theta| < c) = \Pr \left( -c\sqrt{n} < \sqrt{n}(\hat{\Theta} - \Theta) < c\sqrt{n} \right) = 1 - 2\Phi(-c\sqrt{n}) \quad \text{for any } c > 0 \]

\[ \text{SE}(\hat{\Theta}) = \frac{\sigma}{\sqrt{n}} \quad (\sqrt{n} \text{ is size of "typical" distance between } \Theta \text{ and } \hat{\Theta}) \]
In other cases, we might have to use other approaches to learn about the distribution of our estimator.

Suppose now that our population of interest is the bivariate distribution of average LSAT and average GPA for students entering law school across a total of 82 United States law schools.

If we could sample from the population over and over again, we could use simulation to approximate the distribution of $\hat{\Theta}$.

Results from computing $\hat{\Theta}$ for each of 100,000 samples of size $n=15$:

<table>
<thead>
<tr>
<th>Quantiles</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$0.95$</td>
<td>$0.9128$</td>
</tr>
<tr>
<td>$0.75$</td>
<td>$0.8464$</td>
</tr>
<tr>
<td>$0.50$</td>
<td>$0.7699$</td>
</tr>
<tr>
<td>$0.25$</td>
<td>$0.6772$</td>
</tr>
<tr>
<td>$0.05$</td>
<td>$0.5010$</td>
</tr>
</tbody>
</table>

$\min = -0.3647$, $\max = 0.9856$, $\text{Mean} = 0.7464$, $\text{SE} = 0.1302$ (sd of $\hat{\Theta}$ values)

Suppose we were to draw an iid sample from $F$ of size $n=15$ and compute

$$\hat{\Theta} = S(x_1, \ldots, x_n) = \frac{\sum_{i=1}^{15} (y_{i1} - \bar{y}_1)(y_{i2} - \bar{y}_2)}{\sqrt{\sum_{i=1}^{15} (y_{i1} - \bar{y}_1)^2} \sqrt{\sum_{i=1}^{15} (y_{i2} - \bar{y}_2)^2}}$$

where $x_i = [y_{i1}, y_{i2}]$, $y_{i1}$ = average LSAT for $i^{th}$ randomly selected school, $y_{i2}$ = average GPA for $i^{th}$ randomly selected school.

How good would our estimator $\hat{\Theta}$ be?
This works fine if we can sample repeatedly from our population. However, in most real problems, we get only one sample! How can we use our single sample to get information about the distribution of our estimator $\hat{\theta}$?

**Bootstrap Resampling**

Use these to approximate the distribution of $\hat{\theta}$.

$F$ is an iid sample of size $n$ (same as the original sample $F$). Each of these is an iid sample of size $n$ (the estimate of $F$).

**The Simulation Approach**

Use $X_1, \ldots, X_n \sim F$ to estimate $\theta$ by $\hat{\theta} = \frac{1}{n} \sum_{i=1}^{n} X_i$ and $F$ by $\hat{F}$. Then use the simulation approach with $\hat{F}$ in place of $F$ to approximate the distribution of $\hat{\theta}$. 

**The Bootstrap Approach**

Sample 1 $\rightarrow \hat{\theta}^1$
Sample 2 $\rightarrow \hat{\theta}^2$
Sample 3 $\rightarrow \hat{\theta}^3$
Sample B $\rightarrow \hat{\theta}_B$
How should we obtain \( \hat{F} \) to estimate \( F \)?

If we know that our population has some parametric distribution (e.g., bivariate normal), we can estimate the parameters of that distribution using standard methods.

For example, if \( F \) is \( N(\mu, \sigma^2) \) use \( \hat{F} \) \( N(\hat{\mu}, \hat{\sigma}^2) \) where \( \hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} y_i = \bar{y} \), \( \hat{\sigma}^2 = \frac{1}{n-1} \sum_{i=1}^{n} (y_i - \bar{y})^2 \).

If we don't know anything about \( F \), we usually take a nonparametric approach and use the Empirical Cumulative Distribution Function (ECDF) to approximate \( F \).

Sampling from this distribution is equivalent to drawing with-replacement simple random samples from the observed sample \( y_1, y_2, \ldots, y_n \).

The Empirical Cumulative Distribution Function (ECDF) assigns probability \( \frac{1}{n} \) to each \( y_i \) (i = 1, ..., n).

For example, suppose our sample is 3.4, 1.2, 8.5, 7.6, 4.9. Then \( \hat{F} \) assigns probability \( \frac{1}{5} \) to each of 1.2, 3.4, 4.9, 7.6, and 8.5.

\[ \hat{F}(y) = \frac{3}{5} = 0.4 \]

\[ \hat{F}(4.9) = \frac{3}{5} = 0.6 \]
As an example, let's see how the bootstrap works for the law school example.

The goal there is to estimate the correlation between average LSAT and average GPA in the population of 82 law schools using a sample of 15 law schools.

We also want to be able to learn about the distribution of our estimator using the sample of 15 law schools.

#Install the packages bootstrap and boot.
#Load the bootstrap package.
library(bootstrap)

#This package is only loaded so that the law school data sets can be accessed.

#The entire population of 82 law schools.
law82
<table>
<thead>
<tr>
<th>Rank</th>
<th>Law School</th>
<th>LSAT</th>
<th>GPA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>576</td>
<td>339</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>635</td>
<td>330</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>558</td>
<td>281</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>578</td>
<td>303</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>666</td>
<td>344</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>580</td>
<td>307</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>555</td>
<td>300</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>661</td>
<td>343</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>651</td>
<td>336</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>605</td>
<td>313</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>653</td>
<td>312</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>575</td>
<td>274</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td>545</td>
<td>276</td>
</tr>
<tr>
<td>14</td>
<td></td>
<td>572</td>
<td>288</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>594</td>
<td>296</td>
</tr>
</tbody>
</table>

The random sample of 15 law schools.
# Some example bootstrap samples.

```r
set.seed(511)

law[sample(1:15,replace=T),]
```

<table>
<thead>
<tr>
<th>LSAT</th>
<th>GPA</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>661</td>
</tr>
<tr>
<td>3</td>
<td>558</td>
</tr>
<tr>
<td>11</td>
<td>653</td>
</tr>
<tr>
<td>12</td>
<td>575</td>
</tr>
<tr>
<td>5</td>
<td>666</td>
</tr>
<tr>
<td>15</td>
<td>594</td>
</tr>
<tr>
<td>5.1</td>
<td>666</td>
</tr>
<tr>
<td>6</td>
<td>580</td>
</tr>
<tr>
<td>3.1</td>
<td>558</td>
</tr>
<tr>
<td>13</td>
<td>545</td>
</tr>
<tr>
<td>9</td>
<td>651</td>
</tr>
<tr>
<td>2</td>
<td>635</td>
</tr>
<tr>
<td>5.2</td>
<td>666</td>
</tr>
<tr>
<td>3.2</td>
<td>558</td>
</tr>
<tr>
<td>8.1</td>
<td>661</td>
</tr>
</tbody>
</table>

```r
```
```r

law[sample(1:15,replace=T),]
```

<table>
<thead>
<tr>
<th>LSAT</th>
<th>GPA</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>572</td>
</tr>
<tr>
<td>4</td>
<td>578</td>
</tr>
<tr>
<td>15</td>
<td>594</td>
</tr>
<tr>
<td>1</td>
<td>576</td>
</tr>
<tr>
<td>15.1</td>
<td>594</td>
</tr>
<tr>
<td>5</td>
<td>666</td>
</tr>
<tr>
<td>14.1</td>
<td>572</td>
</tr>
<tr>
<td>4.1</td>
<td>578</td>
</tr>
<tr>
<td>5.1</td>
<td>666</td>
</tr>
<tr>
<td>1.1</td>
<td>576</td>
</tr>
<tr>
<td>3.1</td>
<td>558</td>
</tr>
<tr>
<td>15.2</td>
<td>594</td>
</tr>
<tr>
<td>6</td>
<td>580</td>
</tr>
</tbody>
</table>

```r
```
```r

# Load the boot package.

```r
library(boot)
```

```r
# Write a function that will compute the statistic for any bootstrap sample denoted by i=indices of resampled observations.

```r
theta.hat=function(d,i)
{
  cor(d[i,1],d[i,2])
}
```

```r
# Perform bootstrapping using the boot function.

```r
set.seed(9373564)
o=boot(data=law,statistic=theta.hat,R=5000)
```
ORDINARY NONPARAMETRIC BOOTSTRAP

Call:
boot(data=law, statistic=theta.hat, R=5000)

Bootstrap Statistics :
          original    bias    std. error
t1* 0.7763745 -0.006403791   0.1348234

# Examine the sample statistic theta.hat.
# This is the sample correlation coefficient.
o$t0
[1] 0.7763745

# Examine the distribution of any bootstrap
# replications of theta.hat. These are the
# sample correlation coefficients computed
# for each of the 5000 bootstrap samples.
hist(o$t,xlab="theta.hat.star",
    main="Histogram of the 5000 Bootstrap Replications",
    nclass=50,col=4)
box()

# We can examine characteristics of the
# bootstrap distribution for theta.hat.
quantile(o$t,c(.95,.75,.5,.25,.05))
  95%       75%       50%       25%       5%
0.9468013 0.8748430 0.7873167 0.6894694 0.5256462

min(o$t)
[1] 0.02791739

max(o$t)
[1] 0.9946538

mean(o$t)
[1] 0.7699707

sd(o$t)
[1] 0.1348234
#How does this compare to the true distribution \( \theta_{true} \)? If this were a real problem, we wouldn't know for sure. However, in this case, we know the true population and have previously used simulation to get a good approximation of the true distribution of \( \theta_{true} \).

<table>
<thead>
<tr>
<th>Estimates based on</th>
<th>100,000 Simulation</th>
<th>5000 Bootstrap Replications</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.95 Quantiles</td>
<td>0.9128</td>
<td>0.9468</td>
</tr>
<tr>
<td>0.75 Quantiles</td>
<td>0.8404</td>
<td>0.8748</td>
</tr>
<tr>
<td>0.50 Quantiles</td>
<td>0.7699</td>
<td>0.7873</td>
</tr>
<tr>
<td>0.25 Quantiles</td>
<td>0.6777</td>
<td>0.6895</td>
</tr>
<tr>
<td>0.05 Quantiles</td>
<td>0.5010</td>
<td>0.5256</td>
</tr>
<tr>
<td>Min</td>
<td>-0.3647</td>
<td>0.0279</td>
</tr>
<tr>
<td>Max</td>
<td>0.9856</td>
<td>0.9947</td>
</tr>
<tr>
<td>Mean</td>
<td>0.7464</td>
<td>0.7700</td>
</tr>
<tr>
<td>SE</td>
<td>0.1302</td>
<td>0.1348</td>
</tr>
</tbody>
</table>

Comparing back to the simulation results, we see that the bootstrap distribution seems shifted to the right relative to the true distribution. This is perhaps not surprising in this case, given that our sample estimate \( \theta_{true} \) is an overestimate of the population parameter \( \theta \).

```r
cor(law82[,2],law82[,3])
```

However, note that the standard error estimate is pretty good (0.1348 compared to 0.1302).
The bias of $\theta_{\text{hat}}$ is estimated to be

$$\text{mean}(o\$t) - o\$t_0$$

[1] -0.006403791

The bias corrected estimate is

$$o\$t_0 - (\text{mean}(o\$t) - o\$t_0)$$

[1] 0.7827783

However, note that our previous simulation did demonstrate that the bias of $\theta_{\text{hat}}$ is negative.

The mean of 100,000 replications of $\theta_{\text{hat}}$ was 0.7464. Thus, the true bias is approximated by

$$0.7464 - \text{cor(law82[,2],law82[,3])}$$

[1] -0.01359786

Thus, the bootstrap estimate of bias was in the right direction even if correcting for it was harmful in this case.

Bias correction can be dangerous in practice. $\hat{\theta}$ may be substantially more variable than $\tilde{\theta}$.

Thus, $\text{MSE}(\hat{\theta}) = E((\hat{\theta} - \theta)^2)$

$= \text{Var}(\hat{\theta}) + [\text{Bias}(\hat{\theta})]^2$

May be larger than

$\text{MSE}(\tilde{\theta}) = \text{Var}(\tilde{\theta}) + [\text{Bias}(\tilde{\theta})]^2$
The Percentile Bootstrap Confidence Interval:

Goal is to obtain an approximate $100(1-\alpha)$% confidence interval for $\theta$.

Let $\hat{\theta}_L = 100(\alpha/2)$ percentile of the bootstrap distribution of $\hat{\theta}$.

Let $\hat{\theta}_U = 100(1-\alpha/2)$ percentile of the bootstrap distribution of $\hat{\theta}$.

Then $[\hat{\theta}_L^*, \hat{\theta}_U^*]$ is an approximate $100(1-\alpha)$% confidence interval for $\theta$.

This interval is called the percentile bootstrap confidence interval for $\theta$.

#Let's find an approximate 95% confidence interval for theta using the percentile bootstrap approach for the law data.

#o is the output of a call to the function boot.

boot.ci(o,conf=.95,type="perc")

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 5000 bootstrap replicates

Intervals :
Level  Percentile
95%  ( 0.4602, 0.9604 )

Calculations and Intervals on Original Scale

quantile(o$t,c(0.025,0.975))

2.5%    97.5%
0.4606444 0.9603306

#The above results don't exactly agree because different algorithms are used for finding the quantiles. These differences are not important.

#Let's plot the interval on the histogram of bootstrap replications of theta.hat.

abline(v=quantile(o$t,c(0.025,0.975)),col=2,lwd=2)
BCa intervals:

BCa stands for Bias-Corrected and Accelerated percentile bootstrap confidence intervals.

The BCa intervals are just like the percentile intervals except that different percentiles of the bootstrap distribution are used.

The percentile bootstrap confidence interval covers $\theta$ with probability less than $1-\alpha$ when $n$ is small in many cases.

Coverage improves for large $n$, but it is possible to develop better intervals by using adjusted percentiles as follows.

In particular, the $100(\alpha_1)$ and $100(\alpha_2)$ percentiles are used, where

$$\alpha_1 = \Phi \left( \frac{\hat{Z}_0 + \frac{\hat{Z}_0 + Z^{(1/2)}}{1 - \hat{A} (\hat{Z}_0 + Z^{(1/2)})}}{} \right)$$

$$\alpha_2 = \Phi \left( \frac{\hat{Z}_0 + \frac{\hat{Z}_0 + Z^{(1-\alpha_2)}}{1 - \hat{A} (\hat{Z}_0 + Z^{(1-\alpha_2)})}}{} \right)$$

$$\hat{A} = \text{"acceleration"} = \frac{\sum_{i=1}^{n} (\hat{\theta}_i - \hat{\theta}_c)^3}{\sum_{i=1}^{n} (\hat{\theta}_i - \hat{\theta}_c)^2 \, \hat{\theta}_c^{3/2}}$$

"rate of change of $\text{SE}(\hat{\theta})$ with respect to $\theta$, measured on a normalized scale"
# Now let's find the BCa interval using the law data.

```
BCa.int.boot.ci(o, conf=.95, type="bca")
```

```
BCa.int

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 5000 bootstrap replicates

Intervals:
Level      BCa
95%     ( 0.3176,  0.9413 )

Calculations and Intervals on Original Scale
```

\[
\bar{Z}_0 = \Phi^{-1}\left(\sum_{b=1}^B \frac{1}{B} \mathbb{1}\{\hat{\theta}^*_b < \hat{\theta}\}\right)
\]

= "A measure of the median bias of \( \hat{\theta}^* \)"

\[
Z(Y) = \Phi^{-1}(Y) \quad (e.g., \Phi^{-1}(0.95) = 1.645, \quad \Phi^{-1}(0.975) = 1.96, \ etc.)
\]

\( \hat{\theta}_i \) = \( \hat{\theta} \) computed without observation \( i \) \( (i=1,...,n) \)

\( \hat{\theta}_- = \frac{1}{n} \sum_{i=1}^{n} \hat{\theta}_i \)

```

\`
attributes(BCa.int)
$names
[1] "R"  "t0"  "call"  "bca"

$class
[1] "bootci"

BCa.int$bca
cnf
[1,] 0.95 27.77 4693.07 0.3175669 0.9413232

sort(o$t)[c(27,28,4693,4694)]
[1] 0.3166779 0.3178323 0.9413132 0.9414529

abline(v=BCa.int$bca[1,4:5],col=3,lwd=2)
```
An Example Involving Smoothing

n=164 men took part in an experiment to see if the drug cholostyramine lowers blood cholesterol levels.

The men in the study were supposed to take six packets of the drug each day, but many of them actually took much less.

The percentage of the intended dose actually taken by each man was determined.

This percentage is a measure of compliance.

The decrease in total blood plasma cholesterol level from the beginning of the experiment to the end of the experiment was also recorded for each man.

The data are available in the R package "bootstrap."

#Install the packages bootstrap and boot.
#Load the bootstrap package.

library(bootstrap)

#This package is only loaded so that the cholostyramine dataset can be accessed.

head(cholost)

z   y
1   0 -5.25
2  27 -1.50
3  71  59.50
4  95  32.50
5   0 -7.25
6  28  23.50

#z is the percentage that represents compliance.
#y is the decrease in total blood plasma cholesterol level from the beginning of the experiment to the end.
#Load the SemiPar package.
library(SemiPar)

#Fit a penalized linear spline to the data. Use the REML approach to determine the smoothing parameter.

y = cholost$y
z = cholost$z

o = spm(y ~ f(z, basis = "trunc.poly", degree = 1))

#Add the fitted spline to the scatterplot.
lines(o, shade = F, se = F)
title("Penalized Linear Spline Fit")

#Was the mean improvement significantly greater at compliance equal 50% than at compliance equal 25%?

comp = data.frame(z = c(25, 50))

pred = predict(o, newdata = comp)
pred

[1]  8.679566 29.394166

[1] 20.7146

#Let's find confidence intervals to accompany this prediction.
# Load the boot package.
library(boot)

# Write a function that will compute the statistic for any bootstrap sample denoted by i=indices of resampled observations.
theta.hat=function(d,i)
{
ystar<-d$y[i]
zstar<-d$z[i]
o=spm(ystar~f(zstar,basis="trunc.poly",degree=1))
pred=predict(o,newdata=compstar)
pred[2]-pred[1]
}

compstar=data.frame(zstar=c(25,50))
compstar
   zstar
1   25
2   50

# Perform bootstrapping using the boot function.
set.seed(90125)
oboos=boot(data=cholost,statistic=theta.hat,R=5000)
oboos

ORDINARY NONPARAMETRIC BOOTSTRAP

Call:
boot(data = cholost, statistic = theta.hat, R = 5000)

Bootstrap Statistics :
        original   bias   std. error
t1*     20.7146 1.297215    8.500598

# Examine the sample statistic theta.hat. This is the difference between predictions at compliance=50% and compliance=25%.
oboos$t0
[1] 20.7146

# Examine the distribution of 5000 bootstrap replications of theta.hat. These are the difference between predictions at 50% and 25% for each of the 5000 bootstrap samples.
hist(oboos$t,xlab="theta.hat.star",main="Histogram of the 5000 Bootstrap Replications",nclass=50,col=4)
box()
# Let's find an approximate 95% confidence interval for theta using the percentile bootstrap approach.

```r
boot.ci(boot.out = oboot, conf = 0.95, type = "perc")
```

```
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 5000 bootstrap replicates

CALL :
boot.ci(boot.out = oboot, conf = 0.95, type = "perc")

Intervals :
Level Percentile
95% ( 9.92, 40.18 )
Calculations and Intervals on Original Scale
```

```r
abline(v = quantile(boot$t, c(0.025, 0.975)), col = 2, lwd = 2)
```

# Now let's find the BCa interval.

```r
BCa.int = boot.ci(boot.out = oboot, conf = 0.95, type = "bca")
```

```
BCa.int
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 5000 bootstrap replicates

CALL :
boot.ci(boot.out = oboot, conf = 0.95, type = "bca")

Intervals :
Level BCa
95% ( 9.56, 39.42 )
Calculations and Intervals on Original Scale
```

```r
abline(v = BCa.int$bc[1, 4:5], col = 3, lwd = 2)
```
An Example of Parametric Bootstrapping in the General Linear Model Framework:

Let \( \mathbf{y} = \mathbf{X}\beta + \mathbf{e} \), \( \mathbf{e} \sim N(\mathbf{0}, \mathbf{\Sigma}) \).

1. Simulate \( \mathbf{y}^* = \mathbf{X}\hat{\beta}_e + \mathbf{e}_e^* \), where \( \mathbf{\Sigma}^* \) and \( \hat{\beta}_e^* \) from \( \mathbf{y}^* \) and \( \mathbf{X} \).

2. Compute \( \tilde{\mathbf{e}}^* = (\mathbf{X}^*\mathbf{X}^*)^{-1}\mathbf{X}^*(\mathbf{y}^* - \mathbf{X}\hat{\beta}_e^*) \).

3. Compute \( t^* = \frac{\mathbf{e}_e^* - C\hat{\beta}_e^*}{\tilde{\mathbf{e}}^*} \).

4. Repeat steps 1 through 3 \( B \) times to obtain \( t_1^*, t_2^*, \ldots, t_B^* \) and denote them by \( t_1, t_2, \ldots, t_B \), respectively.

5. Find the \( \alpha/2 \) and \( 1 - \alpha/2 \) quantiles of \( t_1, t_2, \ldots, t_B \) and \( t_1^*, t_2^*, \ldots, t_B^* \) and denote them by \( t_{(1)} \) and \( t_{(B)} \), respectively.

Suppose we wish to find a confidence interval for an estimable function \( \mathbf{s}'\beta \).

We estimate \( \mathbf{s}'\beta \) by \( \mathbf{s}'\hat{\beta}_e = \mathbf{s}'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} \).
Recall the seedling dry weight example.

Trays play the role of experimental units with genotypes as treatments.

Seedlings within trays were the observational units.

The number of observational units varied across experimental units.

This sort of imbalance causes the linear mixed effects model inference to be approximate.

The code below illustrates one way that parametric bootstrap methods can be used to obtain approximate inference.
```r
# Load the data

d = read.delim("http://www.public.iastate.edu/~dnett/SS11/SeedlingDryWeight2.txt")

# Display the data

d
```

<table>
<thead>
<tr>
<th>Genotype</th>
<th>Tray</th>
<th>Seedling</th>
<th>SeedlingWeight</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>A</td>
<td>2</td>
<td>2</td>
<td>9</td>
</tr>
<tr>
<td>A</td>
<td>3</td>
<td>3</td>
<td>11</td>
</tr>
<tr>
<td>A</td>
<td>4</td>
<td>4</td>
<td>12</td>
</tr>
<tr>
<td>A</td>
<td>5</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>A</td>
<td>6</td>
<td>1</td>
<td>17</td>
</tr>
<tr>
<td>A</td>
<td>7</td>
<td>2</td>
<td>17</td>
</tr>
<tr>
<td>A</td>
<td>8</td>
<td>3</td>
<td>16</td>
</tr>
<tr>
<td>A</td>
<td>9</td>
<td>4</td>
<td>15</td>
</tr>
<tr>
<td>A</td>
<td>10</td>
<td>2</td>
<td>19</td>
</tr>
<tr>
<td>A</td>
<td>11</td>
<td>3</td>
<td>18</td>
</tr>
<tr>
<td>A</td>
<td>12</td>
<td>4</td>
<td>18</td>
</tr>
<tr>
<td>A</td>
<td>13</td>
<td>5</td>
<td>18</td>
</tr>
<tr>
<td>A</td>
<td>14</td>
<td>6</td>
<td>24</td>
</tr>
<tr>
<td>A</td>
<td>15</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>A</td>
<td>16</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>A</td>
<td>17</td>
<td>3</td>
<td>16</td>
</tr>
<tr>
<td>B</td>
<td>18</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>B</td>
<td>19</td>
<td>5</td>
<td>15</td>
</tr>
<tr>
<td>B</td>
<td>20</td>
<td>6</td>
<td>14</td>
</tr>
<tr>
<td>B</td>
<td>21</td>
<td>4</td>
<td>17</td>
</tr>
<tr>
<td>B</td>
<td>22</td>
<td>2</td>
<td>20</td>
</tr>
<tr>
<td>B</td>
<td>23</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>B</td>
<td>24</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>B</td>
<td>25</td>
<td>5</td>
<td>19</td>
</tr>
<tr>
<td>B</td>
<td>26</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>B</td>
<td>27</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>B</td>
<td>28</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>B</td>
<td>29</td>
<td>4</td>
<td>9</td>
</tr>
<tr>
<td>B</td>
<td>30</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>31</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>B</td>
<td>32</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>B</td>
<td>33</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>B</td>
<td>34</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>B</td>
<td>35</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>B</td>
<td>36</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>37</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>B</td>
<td>38</td>
<td>6</td>
<td>3</td>
</tr>
</tbody>
</table>

---
plot(d[,2],d[,4]+rnorm(56,0,.2),
  xlab="Tray",ylab="Seedling Dry Weight",
  col=2*(1+(d[,1]="B")),pch="-",cex=2)
legend("topright",c("Genotype A","Genotype B"),
  fill=c(2,4),border=c(2,4))

Tray=as.factor(d$Tray)
Geno=as.factor(d$Genotype)
y=d$SeedlingWeight

#Load the lme4 package.
library(lme4)

#Fit a linear mixed effects model with
#fixed effects for genotypes
#and random effects for trays.
o=lmer(y~Geno+(1|Tray))
Summary of linear mixed model fit by REML:

Formula: y ~ Geno + (1 | Tray)

AIC: 255.1
BIC: 263.2
logLik: -123.6
deviance: 253
REMLdev: 247.1

Random effects:
Groups Name Variance Std.Dev.
Tray (Intercept) 11.6612 3.4149
Residual 3.5428 1.8822
Number of obs: 56, groups: Tray, 8

Fixed effects:

<table>
<thead>
<tr>
<th>Estimate</th>
<th>Std. Error</th>
<th>t value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Intercept)</td>
<td>15.289</td>
<td>1.745</td>
</tr>
<tr>
<td>GenoB</td>
<td>-3.550</td>
<td>2.468</td>
</tr>
</tbody>
</table>

Correlation of Fixed Effects:
(Intr)  GenoB -0.707

# Note that lmer computes t-statistics but reports no p-values.

beta.hat = fixef(o)

beta.hat

(Intercept) GenoB
15.288843 -3.550208

se = sqrt(diag(vcov(o)))

se

[1] 1.744965 2.468365

# We can use the simulate function to generate a new y vector by simulating from the fitted linear mixed effects model.

so = simulate(o)

so

[,1]
[1,] 16.980090
[2,] 15.997319
[3,] 17.751943
[4,] 18.403980
[5,] 17.886042
[6,] 19.650497
[7,] 20.261284
[8,] 17.709388
[9,] 17.996296
[10,] 18.720326
[11,] 16.850939
[12,] 19.666343
[13,] 22.707095
[14,] 19.929449
[15,] 15.774276
[16,] 17.701643
[17,] 20.021268
[18,] 17.871258
[19,] 17.371802
[20,] 18.948416
[21,] 18.254235
[22,] 14.883499
[23,] 15.108590
[24,] 15.580144
[25,] 12.252397
[26,] 13.662142
[27,] 12.438441
[28,] 18.494325
[29,] 15.165190
[30,] 11.694548
[31,] 12.168397
[32,] 9.794812
[33,] 12.384336
[34,] 16.698297
[35,] 15.207734
Now let's use parametric bootstrap to approximate the distribution of 
$t = (\text{estimate} - \text{mean})/\text{se(estimate)}$.

In this case, we will focus on estimation of $\beta_2 = \text{GenoB}$, which represents difference between the genotype means.

```r
B = 10000

for(b in 1:B){
  ystar = drop(simulate(o))
  ostar = lmer(ystar ~ Geno + (1|Tray))
  tstar[b] = (fixef(ostar)[2] - beta.hat[2]) / sqrt(vcov(ostar)[2,2])
}
```

Let's take a look at the distribution of bootstrap replications of $t$.

```r
hist(tstar, probability = T, nclass = 50, col = 4)
box()
```
```
tquant=quantile(tstar,c(.025,.975))
tquant
  2.5%   97.5%
-2.459130  2.452459

#Lower endpoint of bootstrap confidence interval
  GenoB
  -9.603771

#Upper endpoint of bootstrap confidence interval
beta.hat[2]-tquant[1]*se[2]
  GenoB
  2.519821
```

```
#lme uses a t with 6 d.f. to approximate
#the distribution of t.
#Is that a good approximation in this case
#according to the bootstrap?

tt=seq(-8,8,by=.01)
lines(tt,dt(tt,6),col=2,lwd=2)
```

```
#The bootstrap distribution of t looks very similar
#to a t distribution with 6 degrees of freedom.
#However, such figures don't necessarily show
#whether there is good agreement in the tails of the
#distribution. Let's check the relevant quantiles.

qt(c(.025,.975),6)
[1] -2.446912  2.446912

tquant=quantile(tstar,c(.025,.975))
tquant
  2.5%   97.5%
-2.459130  2.452459

#The agreement is excellent in this case.
```
Concluding Remarks

In order for the bootstrap to have a good chance of working well, \( \hat{F} \) needs to be a good estimator of \( F \).

When \( Y_1, \ldots, Y_n \sim F \) and \( \hat{F}_n \) is the empirical cumulative distribution function, \( \hat{F}_n \) is guaranteed to be close to \( F \) for large \( n \).

How large should \( n \) be in order for \( \hat{F}_n \) to be a good estimator of \( F \)?

That depends on \( F \).

More formally, if observations are \( k \times 1 \),

\[
\sup_{x \in \mathbb{R}} |\hat{F}_n(x) - F(x)| \overset{a.s.}{\to} 0 \text{ as } n \to \infty.
\]

This is a Stat 642 or 643 result.

The Strong Law of Large Numbers implies

\[
\hat{F}_n(x) \overset{a.s.}{\to} F(x) \text{ as } n \to \infty
\]

for any \( x \). This is a Stat 542 or 543 result.

There's a lot more to learn about the bootstrap.

The books below provide good starting points.
