An Introduction to the Bootstrap
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 $\mathbb{R}$. 
For example, $\Theta = t(F)$ could be the

- mean,
- variance,
- standard deviation,
- coefficient of variation ($\frac{SD}{|\text{MEAN}|}$),
- median or another quantile of a univariate distribution,
- correlation between components of a bivariate distribution,
- etc.
Statistics involves the estimation of population parameters like \( \theta = t(F) \) using sample statistics.

Suppose \( Y_1, Y_2, \ldots, Y_n \) iid \( F \).

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 \( \Theta = t(F) \) is the mean of a univariate distribution defined by \( F \), we might choose

\[
\hat{\Theta} = s(y_1, \ldots, y_n) = \frac{1}{n} \sum_{i=1}^{n} y_i = \overline{y}.
\]

as our estimator of \( \Theta \).
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 \( \hat{\theta} \)?
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 $N(\theta, 1)$ distribution where $\theta \in \mathbb{R}$ is unknown. Then we know that $\hat{\theta} = \overline{X} \sim N(\theta, 1/n)$.
\[ \hat{\Theta} = \overline{Y}, \sim N(\Theta, \frac{1}{n}) \implies \]

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

\[ \text{Var}(\hat{\Theta}) = \frac{1}{n} \quad (\text{Variance of } \hat{\Theta} \text{ is low for large } n) \]

\[ \Pr( |\hat{\Theta} - \Theta| < c) = \Pr( -c\sqrt{n} \leq \sqrt{n}(\hat{\Theta} - \Theta) \leq c\sqrt{n} ) \]

\[ = 1 - 2\Phi(-c\sqrt{n}) \quad \text{for any } c > 0 \]

\[ SE(\hat{\Theta}) = \frac{\sqrt{n}}{n} \quad (\frac{1}{\sqrt{n}} \text{ is size of "typical" distance between } \Theta \text{ and } \hat{\Theta}) \]
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.
In this case, $F$ is the discrete distribution that assigns probability $1/82$ to each of the $82$ (average LSAT, average GPA) pairs.

Suppose $\Theta = \ell(F)$ is the correlation between average LSAT and average GPA.
Suppose we were to draw an iid sample from $F$ of size $n = 15$ and compute

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

$$
\sqrt{\frac{\sum_{i=1}^{15} (y_{i1} - \overline{y}_{1,1})^2}{\sum_{i=1}^{15} (y_{i2} - \overline{y}_{1,2})^2}}
$$

Where $x_i = [y_{i1}, y_{i2}]'$

$$
= \left[ \begin{array}{c}
\text{average LSAT for } i^{th} \text{ randomly selected school} \\
\text{average GPA for } i^{th} \text{ randomly selected school}
\end{array} \right]
$$

How good would our estimator $\hat{\Theta}$ be?
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>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.95</td>
<td>0.9128</td>
</tr>
<tr>
<td>0.75</td>
<td>0.8404</td>
</tr>
<tr>
<td>0.50</td>
<td>0.7699</td>
</tr>
<tr>
<td>0.25</td>
<td>0.6777</td>
</tr>
<tr>
<td>0.05</td>
<td>0.5010</td>
</tr>
</tbody>
</table>

- Min = -0.3647
- Max = 0.9856
- Mean = 0.7464
- $SE = 0.1302$ (sd of $\hat{\theta}$ values)
The Simulation Approach
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 $\hat{\theta}$?
The Bootstrap Approach

Use $x_1, \ldots, x_n \overset{iid}{\sim} F$ to estimate

$\theta$ by $\hat{\theta} = S(x_1, \ldots, x_n)$ and

$F$ by $\hat{F}$.

Then use the simulation approach with

$\hat{F}$ in place of $F$ to approximate the distribution of $\hat{\theta}$. 
Bootstrap Resampling

Each of these is an iid sample of size $n$ (same as the original sample size) from $\hat{F}$ (the estimate of $F$).

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

$\Rightarrow$ bootstrap sample 1 $\to$ $\hat{\theta}_1^*$

$\Rightarrow$ bootstrap sample 2 $\to$ $\hat{\theta}_2^*$

$\Rightarrow$ bootstrap sample 3 $\to$ $\hat{\theta}_3^*$

$\Rightarrow$ bootstrap sample $B$ $\to$ $\hat{\theta}_B^*$
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 \( \hat{\sigma} \)

\[
N(\hat{\mu}, \hat{\sigma}^2) \quad \text{where} \quad \hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} y_i = \bar{y}, \quad \hat{\sigma}^2 = \frac{1}{n-1} \sum_{i=1}^{n} (y_i - \bar{y})^2.
\]
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 $x_1, x_2, \ldots, x_n$. 
The Empirical Cumulative Distribution Function:

\[ \hat{F}(x) = \frac{1}{n} \sum_{i=1}^{n} 1(x_i \leq x) \]

where \( 1(x_i \leq x) \) is 1 if each component of \( x_i \) is less than or equal to each component of \( x \).
The Empirical Cumulative Distribution Function (ECDF) assigns probability \( \frac{1}{n} \) to each \( Y_i \) \((i=1, \ldots, n)\).

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.

\[
\hat{F}(4) = \frac{2}{5} = 0.4 \\
\hat{F}(4.9) = \frac{3}{5} = 0.6
\]
If our sample is 3.4, 5.6, 3.4, 7.9, 8.6 then \( \hat{F} \) assigns probability \( \frac{1}{5} \) to each of 5.6, 7.9, and 8.6. The value 3.4 gets probability \( \frac{2}{5} \) because it occurs twice in the sample.

\[
\begin{align*}
\hat{F}(1.5) &= 0 \\
\hat{F}(3.4) &= \frac{2}{5} \\
\hat{F}(5.5) &= \frac{2}{5} \\
\hat{F}(5.6) &= \frac{3}{5} \\
\hat{F}(8) &= \frac{4}{5} \\
\hat{F}(9) &= 1
\end{align*}
\]
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>School</th>
<th>LSAT</th>
<th>GPA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>622 3.23</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>542 2.83</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>579 3.24</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>653 3.12</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>606 3.09</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>576 3.39</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>620 3.10</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>615 3.40</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>553 2.97</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>607 2.91</td>
</tr>
<tr>
<td>11</td>
<td>11</td>
<td>558 3.11</td>
</tr>
<tr>
<td>12</td>
<td>12</td>
<td>596 3.24</td>
</tr>
<tr>
<td>13</td>
<td>13</td>
<td>635 3.30</td>
</tr>
<tr>
<td>14</td>
<td>14</td>
<td>581 3.22</td>
</tr>
<tr>
<td>15</td>
<td>15</td>
<td>661 3.43</td>
</tr>
<tr>
<td>16</td>
<td>16</td>
<td>547 2.91</td>
</tr>
<tr>
<td>17</td>
<td>17</td>
<td>599 3.23</td>
</tr>
<tr>
<td>18</td>
<td>18</td>
<td>646 3.47</td>
</tr>
<tr>
<td>19</td>
<td>19</td>
<td>622 3.15</td>
</tr>
<tr>
<td>20</td>
<td>20</td>
<td>611 3.33</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>------</td>
</tr>
<tr>
<td>21</td>
<td>21</td>
<td>546</td>
</tr>
<tr>
<td>22</td>
<td>22</td>
<td>614</td>
</tr>
<tr>
<td>23</td>
<td>23</td>
<td>628</td>
</tr>
<tr>
<td>24</td>
<td>24</td>
<td>575</td>
</tr>
<tr>
<td>25</td>
<td>25</td>
<td>662</td>
</tr>
<tr>
<td>26</td>
<td>26</td>
<td>627</td>
</tr>
<tr>
<td>27</td>
<td>27</td>
<td>608</td>
</tr>
<tr>
<td>28</td>
<td>28</td>
<td>632</td>
</tr>
<tr>
<td>29</td>
<td>29</td>
<td>587</td>
</tr>
<tr>
<td>30</td>
<td>30</td>
<td>581</td>
</tr>
<tr>
<td>31</td>
<td>31</td>
<td>605</td>
</tr>
<tr>
<td>32</td>
<td>32</td>
<td>704</td>
</tr>
<tr>
<td>33</td>
<td>33</td>
<td>477</td>
</tr>
<tr>
<td>34</td>
<td>34</td>
<td>591</td>
</tr>
<tr>
<td>35</td>
<td>35</td>
<td>578</td>
</tr>
<tr>
<td>36</td>
<td>36</td>
<td>572</td>
</tr>
<tr>
<td>37</td>
<td>37</td>
<td>615</td>
</tr>
<tr>
<td>38</td>
<td>38</td>
<td>606</td>
</tr>
<tr>
<td>39</td>
<td>39</td>
<td>603</td>
</tr>
<tr>
<td>40</td>
<td>40</td>
<td>535</td>
</tr>
<tr>
<td>41</td>
<td>41</td>
<td>595</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>42</td>
<td>42</td>
<td>575</td>
</tr>
<tr>
<td>43</td>
<td>43</td>
<td>573</td>
</tr>
<tr>
<td>44</td>
<td>44</td>
<td>644</td>
</tr>
<tr>
<td>45</td>
<td>45</td>
<td>545</td>
</tr>
<tr>
<td>46</td>
<td>46</td>
<td>645</td>
</tr>
<tr>
<td>47</td>
<td>47</td>
<td>651</td>
</tr>
<tr>
<td>48</td>
<td>48</td>
<td>562</td>
</tr>
<tr>
<td>49</td>
<td>49</td>
<td>609</td>
</tr>
<tr>
<td>50</td>
<td>50</td>
<td>555</td>
</tr>
<tr>
<td>51</td>
<td>51</td>
<td>586</td>
</tr>
<tr>
<td>52</td>
<td>52</td>
<td>580</td>
</tr>
<tr>
<td>53</td>
<td>53</td>
<td>594</td>
</tr>
<tr>
<td>54</td>
<td>54</td>
<td>594</td>
</tr>
<tr>
<td>55</td>
<td>55</td>
<td>560</td>
</tr>
<tr>
<td>56</td>
<td>56</td>
<td>641</td>
</tr>
<tr>
<td>57</td>
<td>57</td>
<td>512</td>
</tr>
<tr>
<td>58</td>
<td>58</td>
<td>631</td>
</tr>
<tr>
<td>59</td>
<td>59</td>
<td>597</td>
</tr>
<tr>
<td>60</td>
<td>60</td>
<td>621</td>
</tr>
<tr>
<td>61</td>
<td>61</td>
<td>617</td>
</tr>
<tr>
<td>62</td>
<td>62</td>
<td>637</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>63</td>
<td>62</td>
<td>572</td>
</tr>
<tr>
<td>64</td>
<td>64</td>
<td>610</td>
</tr>
<tr>
<td>65</td>
<td>65</td>
<td>562</td>
</tr>
<tr>
<td>66</td>
<td>66</td>
<td>635</td>
</tr>
<tr>
<td>67</td>
<td>67</td>
<td>614</td>
</tr>
<tr>
<td>68</td>
<td>68</td>
<td>546</td>
</tr>
<tr>
<td>69</td>
<td>69</td>
<td>598</td>
</tr>
<tr>
<td>70</td>
<td>70</td>
<td>666</td>
</tr>
<tr>
<td>71</td>
<td>71</td>
<td>570</td>
</tr>
<tr>
<td>72</td>
<td>72</td>
<td>570</td>
</tr>
<tr>
<td>73</td>
<td>73</td>
<td>605</td>
</tr>
<tr>
<td>74</td>
<td>74</td>
<td>565</td>
</tr>
<tr>
<td>75</td>
<td>75</td>
<td>686</td>
</tr>
<tr>
<td>76</td>
<td>76</td>
<td>608</td>
</tr>
<tr>
<td>77</td>
<td>77</td>
<td>595</td>
</tr>
<tr>
<td>78</td>
<td>78</td>
<td>590</td>
</tr>
<tr>
<td>79</td>
<td>79</td>
<td>558</td>
</tr>
<tr>
<td>80</td>
<td>80</td>
<td>611</td>
</tr>
<tr>
<td>81</td>
<td>81</td>
<td>564</td>
</tr>
<tr>
<td>82</td>
<td>82</td>
<td>575</td>
</tr>
</tbody>
</table>
#The random sample of 15 law schools.

<table>
<thead>
<tr>
<th></th>
<th>LSAT</th>
<th>GPA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>576</td>
<td>339</td>
</tr>
<tr>
<td>2</td>
<td>635</td>
<td>330</td>
</tr>
<tr>
<td>3</td>
<td>558</td>
<td>281</td>
</tr>
<tr>
<td>4</td>
<td>578</td>
<td>303</td>
</tr>
<tr>
<td>5</td>
<td>666</td>
<td>344</td>
</tr>
<tr>
<td>6</td>
<td>580</td>
<td>307</td>
</tr>
<tr>
<td>7</td>
<td>555</td>
<td>300</td>
</tr>
<tr>
<td>8</td>
<td>661</td>
<td>343</td>
</tr>
<tr>
<td>9</td>
<td>651</td>
<td>336</td>
</tr>
<tr>
<td>10</td>
<td>605</td>
<td>313</td>
</tr>
<tr>
<td>11</td>
<td>653</td>
<td>312</td>
</tr>
<tr>
<td>12</td>
<td>575</td>
<td>274</td>
</tr>
<tr>
<td>13</td>
<td>545</td>
<td>276</td>
</tr>
<tr>
<td>14</td>
<td>572</td>
<td>288</td>
</tr>
<tr>
<td>15</td>
<td>594</td>
<td>296</td>
</tr>
</tbody>
</table>
Some example bootstrap samples.

```r
set.seed(511)

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

    LSAT  GPA
 8   661 343
 3   558 281
 11  653 312
 12  575 274
  5  666 344
 15  594 296
  5.1 666 344
  6  580 307
  3.1 558 281
 13  545 276
  9  651 336
  2  635 330
  2  558 281
  5.2 666 344
  3.2 558 281
  8.1 661 343
```
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>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
law[sample(1:15, replace=T),]

<table>
<thead>
<tr>
<th>LSAT</th>
<th>GPA</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>594</td>
</tr>
<tr>
<td>6</td>
<td>580</td>
</tr>
<tr>
<td>12</td>
<td>575</td>
</tr>
<tr>
<td>9</td>
<td>651</td>
</tr>
<tr>
<td>12.1</td>
<td>575</td>
</tr>
<tr>
<td>10</td>
<td>605</td>
</tr>
<tr>
<td>7</td>
<td>555</td>
</tr>
<tr>
<td>10.1</td>
<td>605</td>
</tr>
<tr>
<td>13</td>
<td>545</td>
</tr>
<tr>
<td>13.1</td>
<td>545</td>
</tr>
<tr>
<td>15.1</td>
<td>594</td>
</tr>
<tr>
<td>13.2</td>
<td>545</td>
</tr>
<tr>
<td>1</td>
<td>576</td>
</tr>
<tr>
<td>14</td>
<td>572</td>
</tr>
<tr>
<td>15.2</td>
<td>594</td>
</tr>
</tbody>
</table>
```
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) {
  cor(d[i,1],d[i,2])
}

#Perform bootstrapping using the boot function.

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 :

<table>
<thead>
<tr>
<th></th>
<th>original</th>
<th>bias</th>
<th>std. error</th>
</tr>
</thead>
<tbody>
<tr>
<td>t1*</td>
<td>0.7763745</td>
<td>-0.006403791</td>
<td>0.1348234</td>
</tr>
</tbody>
</table>
#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 \( \theta \).

\[
\text{quantile}(o \times t, c(0.95, 0.75, 0.5, 0.25, 0.05))
\]

<table>
<thead>
<tr>
<th>95%</th>
<th>75%</th>
<th>50%</th>
<th>25%</th>
<th>5%</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9468013</td>
<td>0.8748430</td>
<td>0.7873167</td>
<td>0.6894694</td>
<td>0.5256462</td>
</tr>
</tbody>
</table>

\[
\text{min}(o \times t)
\]

[1] 0.02791739

\[
\text{max}(o \times t)
\]

[1] 0.9946538

\[
\text{mean}(o \times t)
\]

[1] 0.7699707

\[
\text{sd}(o \times t)
\]

[1] 0.1348234
How does this compare to the true distribution of \( \theta \)? 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 \( \theta \).

<table>
<thead>
<tr>
<th>Estimates based on 100,000 Simulation</th>
<th>Estimates based on 5000 Bootstrap Replications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quantiles</td>
<td></td>
</tr>
<tr>
<td>0.95</td>
<td>0.9128</td>
</tr>
<tr>
<td>0.75</td>
<td>0.8404</td>
</tr>
<tr>
<td>0.50</td>
<td>0.7699</td>
</tr>
<tr>
<td>0.25</td>
<td>0.6777</td>
</tr>
<tr>
<td>0.05</td>
<td>0.5010</td>
</tr>
<tr>
<td>Min</td>
<td>-0.3647</td>
</tr>
<tr>
<td>Max</td>
<td>0.9856</td>
</tr>
<tr>
<td>Mean</td>
<td>0.7464</td>
</tr>
<tr>
<td>SE</td>
<td>0.1302</td>
</tr>
<tr>
<td></td>
<td>0.9468</td>
</tr>
<tr>
<td></td>
<td>0.8748</td>
</tr>
<tr>
<td></td>
<td>0.7873</td>
</tr>
<tr>
<td></td>
<td>0.6895</td>
</tr>
<tr>
<td></td>
<td>0.5256</td>
</tr>
<tr>
<td></td>
<td>0.0279</td>
</tr>
<tr>
<td></td>
<td>0.9947</td>
</tr>
<tr>
<td></td>
<td>0.7700</td>
</tr>
<tr>
<td></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 $\theta_{\hat{}}$ is an overestimate of the population parameter $\theta$.

```
o$\theta_{\hat{}}$
[1] 0.7763745
```

```
cor(law82[,2],law82[,3])
[1] 0.7599979
```

However, note that the standard error estimate is pretty good (0.1348 compared to 0.1302).
Using the Bootstrap to Estimate and Correct for Bias:

$$\text{Bias}(\hat{\theta}) = E_f(\hat{\theta}) - \theta$$

Bootstrap Estimate of Bias

$$\hat{\text{Bias}}(\hat{\theta}) = E_f^\hat{\theta}(\hat{\theta}) - \hat{\theta}$$

$$\approx \frac{1}{B} \sum_{b=1}^{B} \hat{\theta}_b^* - \hat{\theta}.$$

(If $\hat{F} = ECDF$)
Bias Corrected Estimate

\[ \hat{\theta} = \hat{\theta} - \text{Bias} (\hat{\theta}) \]

\[ = \hat{\theta} - \left( \frac{1}{B} \sum_{b=1}^{B} \Theta_{b}^* - \hat{\theta} \right) \]

\[ = 2 \hat{\theta} - \frac{1}{B} \sum_{b=1}^{B} \Theta_{b}^* \]
#The bias of theta.hat is estimated to be

```
mean(o$t)-o$t0
[1]  -0.006403791
```

#The bias corrected estimate is

```
o$t0-(mean(o$t)-o$t0)
[1]   0.7827783
```
In this case, our biased corrected estimate
is further from the true theta than the
uncorrected estimate.

o$t0-(mean(o$t)-o$t0)
[1] 0.7827783

cor(law82[,2],law82[,3])
[1] 0.7599979

o$t0
[1] 0.7763745
However, note that our previous simulation did demonstrate that the bias of \( \theta_{\hat{}} \) is negative.

The mean of 100,000 replications of \( \theta_{\hat{}} \) 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 \( \Theta \).

Thus, \( \text{MSE}(\hat{\Theta}) = E(\hat{\Theta} - \Theta)^2 \)
\[ = \text{Var}(\hat{\Theta}) + [\text{Bias}(\hat{\Theta})]^2 \]

May be larger than

\( \text{MSE}(\hat{\Theta}) = \text{Var}(\hat{\Theta}) + [\text{Bias}(\hat{\Theta})]^2 \).
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 \( [\hat{\theta}^*, \hat{\theta}^*] \) 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)
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.
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.
In particular, the 100(\(\alpha_1\)) and 100(\(\alpha_2\)) percentiles are used, where

\[
\alpha_1 = \Phi \left( \frac{\hat{z}_0 + \hat{Z}(\alpha/2)}{1 - \hat{\alpha} (\hat{z}_0 + \hat{Z}(\alpha/2))} \right),
\]

\[
\alpha_2 = \Phi \left( \frac{\hat{z}_0 + \hat{Z}(1-\alpha/2)}{1 - \hat{\alpha} (\hat{z}_0 + \hat{Z}(1-\alpha/2))} \right),
\]

\[
\hat{\alpha} = \text{"acceleration"} = \sum_{i=1}^{n} (\hat{\theta}_i - \hat{\theta}_i)^3 \left/ \left( \sum_{i=1}^{n} (\hat{\theta}_i - \hat{\theta}_i)^2 \right)^{3/2} \right.
\]

"rate of change of \text{SE} (\hat{\theta}) with respect to \theta, measured on a normalized scale"
\[ \hat{Z}_0 = \Phi^{-1} \left( \sum_{b=1}^{B} 1 \left( \hat{\theta}_b^* < \hat{\theta} \right) \right) \]

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

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

\[ \hat{\theta}_{-i} = \hat{\theta} \text{ computed without observation } i \quad (i=1,\ldots,n) \]

\[ \hat{\theta}_- = \frac{1}{n} \sum_{i=1}^{n} \hat{\theta}_{-i} \]
#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
attributes(BCa.int)

$names
[1] "R" "t0" "call" "bca"

$class
[1] "bootci"

BCa.int$bca

  conf
[1,] 0.95 27.77 4693.07 0.3175669 0.9413232

sort(o$t)

[1] 0.3166779 0.3178323 0.9413132 0.9414529

abline(v=BCa.int$bca[1,4:5], col=3, lwd=2)
Histogram of the 5000 Bootstrap Replications

Frequency

theta.hat.star

0.0 0.2 0.4 0.6 0.8 1.0
An Example Involving Smoothing

n=164 men took part in an experiment to see if the drug cholestyramine 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
#cholestyramine 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.
plot(cholost,xlab="Compliance",
    ylab="Improvement",
    pch=19,col=4)
#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%?

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

<table>
<thead>
<tr>
<th></th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>25</td>
</tr>
<tr>
<td>2</td>
<td>50</td>
</tr>
</tbody>
</table>

```r
pred = predict(o, newdata = comp)
pred
```

```
[1]  8.679566 29.394166
```

```r
```

```
[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.

```r
set.seed(90125)
oboot = boot(data = cholost, statistic = theta.hat, R = 5000)
oboot

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%.

oboott0
[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(oboott0,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(oboot,conf=.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

abline(v=quantile(oboot$t,c(0.025,0.975)),col=2,lwd=2)
```
Histogram of the 5000 Bootstrap Replications
# Now let's find the BCa interval.

BCa.int = boot.ci(oboot, conf=.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

abline(v=BCa.int$bca[1,4:5], col=3, lwd=2)
Histogram of the 5000 Bootstrap Replications

Frequency

theta.hat.star

0 10 20 30 40 50 60

0 100 200 300 400
An Example of Parametric Bootstrapping in the General Linear Model Framework:

\[ Y = X\beta + \varepsilon, \quad \varepsilon \sim N(0, \Sigma) \]

Suppose we wish to find a confidence interval for an estimable function \( \gamma' \beta \).

We estimate \( \gamma' \beta \) by \( \gamma' \hat{\beta} \hat{\varepsilon} = \gamma' (X'\hat{\varepsilon}^{-1}X)^{-1}X'\hat{\varepsilon} \).
In some cases,

\[ t = \frac{\hat{\beta}^\prime \hat{\epsilon} - \xi' \beta}{\sqrt{\xi' (X' \hat{\Sigma}^{-1} X)^{-1} \xi}} \sim t_{df} \]

for known degrees of freedom df.

However, in general, \( \xi' \hat{\beta}^\prime \hat{\epsilon} \) is a nonlinear function of \( \xi \) and the distribution of \( t \) is unknown.
If we believe the model
\[ y = X\beta + \varepsilon, \quad \varepsilon \sim N(0, \Sigma) \]
we might try to approximate the distribution of 
\[ t = \frac{\hat{\beta}' \hat{\Sigma}^{-1} \varepsilon - \varepsilon'}{\sqrt{\varepsilon' (X'X)^{-1} \varepsilon}} \]
using a parametric bootstrap approach.
1. Simulate $y^* = X \hat{\beta}_{\Sigma^*} + \xi^*$, 

where $\xi^* \sim N(0, \Sigma)$

2. Compute $\hat{\Sigma}^*$ and $\hat{\beta}_{\Sigma^*}$ from $y^*$ and $X$ just as $\hat{\Sigma}$ and $\hat{\beta}_{\Sigma}$ were obtained from $y$ and $X$.

3. Compute $t^* = \frac{c' \hat{\beta}_{\Sigma^*} - c' \hat{\beta}_{\Sigma}}{\sqrt{c' (X' \Sigma^{-1} X) c}}$
4. Repeat steps 1 through 3 B times to obtain $t_1^*, t_2^*, \ldots, t_B^*$.

5. Find the $\alpha/2$ and $1-\alpha/2$ quantiles of $t_1^*, t_2^*, \ldots, t_B^*$ and denote them by $t_{(\alpha/2)}$ and $t_{(1-\alpha/2)}$, respectively.
6. If the bootstrap distribution of $t$ (approximated by $t_1^*, t_2^*, \ldots, t_B^*$) is a good approximation to the true distribution of $t$, then

$$\Pr\left( t_{(\alpha/2)}^* \leq \frac{\hat{c}'\hat{\beta}^* \pm C}{\sqrt{c'(X'\hat{\Sigma}^{-1}X)'C}} \leq t_{(1-\alpha/2)}^* \right) \approx 1-\alpha.$$ 

$\Rightarrow$ An approximate $100(1-\alpha)$% confidence interval for $\hat{c}'\hat{\beta}$ is

$$\left( \hat{c}'\hat{\beta}^* - t_{(1-\alpha/2)}^* \sqrt{c'(X'\hat{\Sigma}^{-1}X)'C}, \hat{c}'\hat{\beta}^* - t_{(1-\alpha/2)}^* \sqrt{c'(X'\hat{\Sigma}^{-1}X)'C} \right)$$
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
d = read.delim("http://www.public.iastate.edu/~dnett/S511/SeedlingDryWeight2.txt")
d
<table>
<thead>
<tr>
<th>Genotype</th>
<th>Tray</th>
<th>Seedling</th>
<th>SeedlingWeight</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>A</td>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>A</td>
<td>1</td>
<td>11</td>
</tr>
<tr>
<td>4</td>
<td>A</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>5</td>
<td>A</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>A</td>
<td>2</td>
<td>17</td>
</tr>
<tr>
<td>7</td>
<td>A</td>
<td>2</td>
<td>17</td>
</tr>
<tr>
<td>8</td>
<td>A</td>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>9</td>
<td>A</td>
<td>2</td>
<td>15</td>
</tr>
<tr>
<td>10</td>
<td>A</td>
<td>2</td>
<td>19</td>
</tr>
<tr>
<td>11</td>
<td>A</td>
<td>2</td>
<td>18</td>
</tr>
<tr>
<td>12</td>
<td>A</td>
<td>2</td>
<td>18</td>
</tr>
<tr>
<td>13</td>
<td>A</td>
<td>2</td>
<td>18</td>
</tr>
<tr>
<td>14</td>
<td>A</td>
<td>2</td>
<td>24</td>
</tr>
<tr>
<td>15</td>
<td>A</td>
<td>3</td>
<td>12</td>
</tr>
<tr>
<td>16</td>
<td>A</td>
<td>3</td>
<td>12</td>
</tr>
<tr>
<td>17</td>
<td>A</td>
<td>3</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>18</td>
<td>A</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>19</td>
<td>A</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>20</td>
<td>A</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>21</td>
<td>A</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>22</td>
<td>A</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>23</td>
<td>A</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>24</td>
<td>A</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>25</td>
<td>A</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>26</td>
<td>A</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>27</td>
<td>A</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>28</td>
<td>A</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>29</td>
<td>A</td>
<td>4</td>
<td>9</td>
</tr>
<tr>
<td>30</td>
<td>B</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>31</td>
<td>B</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>32</td>
<td>B</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>33</td>
<td>B</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>34</td>
<td>B</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>35</td>
<td>B</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>36</td>
<td>B</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>37</td>
<td>B</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>38</td>
<td>B</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(o)

Linear mixed model fit by REML
Formula: y ~ Geno + (1 | Tray)
    AIC   BIC logLik deviance REMLdev
255.1 263.2 -123.6      253   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:

  Estimate Std. Error  t value
(Intercept)   15.289      1.745    8.762
GenoB         -3.550      2.468   -1.438

Correlation of Fixed Effects:
     (Intr)
GenoB  -0.707
#Note that lmer computes t-statistics but reports no p-values.

\[
\text{beta.hat} = \text{fixef(o)}
\]

\[
\text{beta.hat}
\]

\[
(\text{Intercept}) \quad \text{GenoB} \\
15.288843 \quad -3.550208
\]

\[
\text{se} = \sqrt{\text{diag(vcov(o))}}
\]

\[
\text{se}
\]

\[
[1] 1.744965 \quad 2.468365
\]
# 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
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>15.774276</td>
</tr>
<tr>
<td>16</td>
<td>17.701643</td>
</tr>
<tr>
<td>17</td>
<td>20.021268</td>
</tr>
<tr>
<td>18</td>
<td>17.871258</td>
</tr>
<tr>
<td>19</td>
<td>17.371802</td>
</tr>
<tr>
<td>20</td>
<td>18.948416</td>
</tr>
<tr>
<td>21</td>
<td>18.254235</td>
</tr>
<tr>
<td>22</td>
<td>14.883499</td>
</tr>
<tr>
<td>23</td>
<td>15.108590</td>
</tr>
<tr>
<td>24</td>
<td>15.580144</td>
</tr>
<tr>
<td>25</td>
<td>12.252397</td>
</tr>
<tr>
<td>26</td>
<td>13.662142</td>
</tr>
<tr>
<td>27</td>
<td>12.438441</td>
</tr>
<tr>
<td>28</td>
<td>18.494325</td>
</tr>
<tr>
<td>29</td>
<td>15.165190</td>
</tr>
<tr>
<td>30</td>
<td>11.694548</td>
</tr>
<tr>
<td>31</td>
<td>12.168397</td>
</tr>
<tr>
<td>32</td>
<td>9.794812</td>
</tr>
<tr>
<td>33</td>
<td>12.384336</td>
</tr>
<tr>
<td>34</td>
<td>16.698297</td>
</tr>
<tr>
<td>35</td>
<td>15.207734</td>
</tr>
</tbody>
</table>
# Now let's use parametric bootstrap to approximate the distribution of
# \( t = \frac{\text{estimate} - \text{mean}}{\text{se(estimate)}} \).

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

\[ B = 10000 \]

\[ t^{\star} = \text{rep}(0, B) \]

\[ \text{set.seed}(781) \]

\[ \text{for}(b \text{ in } 1:B) \{
    \ ystar = \text{drop}(\text{simulate}(o))
    \ ostar = \text{lmer}(ystar \sim \text{Geno} + (1 | \text{Tray}))
    \ tstar[b] = (\text{fixef}(ostar)[2] - \text{beta.hat}[2])/\sqrt{\text{vcov}(ostar)[2,2]} 
\} \]
# Let's take a look at the distribution of bootstrap replications of t.

```
hist(tstar,probability=T,nclass=50,col=4)
box()
```
Histogram of t_star
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.
We can also use a parametric bootstrap strategy to directly test a null hypothesis in the general or generalized linear models framework.

The basic steps are as follows.
1. Compute the test statistic of interest from the original data. Denote this statistic $T$.

This could be an F-statistic, the absolute value of a t-statistic, a likelihood ratio test statistic, a Wald statistic or the absolute value of a Wald statistic, or any other statistic where large values provide evidence against the null hypothesis.
2. Estimate the model parameters under the reduced model implied by the null hypothesis.

3. Simulate a new response vector $y^*$ from the null model with parameter values set to the estimates obtained in step 2.
4. Using the data \( y^* \) in place of \( y \), compute the same test statistic computed in step 1. Let \( T^* \) denote the value of this statistic.

5. Repeat steps 3 and 4 \( B \) times to obtain \( T^*_1, \ldots, T^*_B \).
6. The parametric bootstrap p-value for the test of interest is given by the proportion of $T^*_1, \ldots, T^*_B$ greater than or equal to $T$. 
Let's examine a parametric bootstrap test of the null hypothesis that the two genotype means are equal.

The test statistic of interest is

\[ T = \frac{|\hat{\beta}_2|}{se[2]} \]

GenoB
1.438283

Estimate parameters under the reduced model.

onull=lmer(y~1+(1|Tray))
#Generate $T^*$ values.

B = 10000
Tstar = rep(0, B)
set.seed(3432)
for (b in 1:B) {
    ystar = drop(simulate(onull))
    ostar = lmer(ystar ~ Geno + (1 | Tray))
    Tstar[b] = abs(fixef(ostar)[2]/sqrt(vcov(ostar)[2,2]))
}

#Plot the distribution of $T^*$ with $T$.

hist(Tstar, col = 4)
abline(v = T, col = 2, lwd = 2)
box()
#Compute parametric bootstrap p-value.

```r
mean(Tstar>=T)
```

```
[1] 0.2039
```

#This is very similar to the p-value produced by #lme in this case.

```r
library(nlme)
summary(lme(y~Geno,random=~1|Tray,data=d))
```

**Fixed effects: y ~ Geno**

<table>
<thead>
<tr>
<th>Value</th>
<th>Std. Error</th>
<th>DF</th>
<th>t-value</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Intercept)</td>
<td>15.288838</td>
<td>1.745089</td>
<td>48</td>
<td>8.761067</td>
</tr>
<tr>
<td>GenoB</td>
<td>-3.550201</td>
<td>2.468539</td>
<td>6</td>
<td>-1.438179</td>
</tr>
</tbody>
</table>
Now let us consider a testing problem that involves nonparametric bootstrap resampling.
Suppose

\[ y_{11}, y_{12}, \ldots, y_{1n_1} \sim F_1, \]

where \( F_1 \) is some unknown distribution with mean \( \mu_1 \) and standard deviation \( \sigma_1 \).

Suppose

\[ y_{21}, y_{22}, \ldots, y_{2n_2} \sim F_2, \]

where \( F_2 \) is some unknown distribution with mean \( \mu_2 \) and standard deviation \( \sigma_2 \).
Suppose all observations are independent.

Suppose the goal is to test

\[ H_0: \mu_1 = \mu_2. \]
The absolute value of the two-sample t-statistic in this situation is

\[ T = \frac{|\bar{y}_1 - \bar{y}_2|}{\sqrt{s_1^2/n_1 + s_2^2/n_2}}. \]
The null distribution of this statistic is unknown because we do not know $F_1$ or $F_2$.

Even if we assume that $F_1$ and $F_2$ are normal distributions, the exact distribution of the statistic is unknown.

We can use nonparametric bootstrap resampling to get a p-value as follows.
1. Compute the test statistic $T$ using the original data.

2. Compute

$$z_{ij} = y_{ij} - \bar{y}_i + (\bar{y}_1 + \bar{y}_2)/2$$

for all $i$ and $j$. 

3. Sample $n_1$ values with replacement from

$$z_{11}, z_{12}, \ldots, z_{1n_1}$$

and denote the sampled values by

$$z_{11}^*, z_{12}^*, \ldots, z_{1n_1}^*$$.
Sample $n_2$ values with replacement from

$$Z_{21}, Z_{22}, \ldots, Z_{2n_2}$$

and denote the sampled values by

$$Z_{21}^*, Z_{22}^*, \ldots, Z_{2n_2}^*.$$
4. Recompute $T$ using $z_{ij}^*$ in place of $y_{ij}$ for all $i$ and $j$. Denote the result as $T^*$.

5. Repeat steps 3 and 4 $B$ times to obtain $T^*_1, \ldots, T^*_B$.

6. The bootstrap p-value for the test of interest is given by the proportion of $T^*_1, \ldots, T^*_B$ greater than or equal to $T$. 
# Here is an example of a test procedure using nonparametric bootstrap resampling.

# Generate data from different distributions. Each distribution has a mean and variance.

```r
set.seed(1393)
y1 = 9 + 2 * rt(10, 3)
y2 = rnorm(15, 6, 1)
```

```r
y1
7.8633884  2.8468387  0.7147789 12.0797314  10.7190609  8.5006002  5.4086213  8.7004333 10.2330230   8.6976407
```

```r
y2
```
# Suppose we want to test if the means of the two distributions are equal.

# Compute a test statistic.

T = abs(t.test(y1, y2)$statistic)
T
1.128351

# Compute zij values.

z1 = y1 - mean(y1) + (mean(y1) + mean(y2))/2
z2 = y2 - mean(y2) + (mean(y1) + mean(y2))/2

mean(z1)
[1] 6.915524

mean(z2)
[1] 6.915524
# Generate T* values.

B = 10000
Tstar = rep(0, B)
set.seed(2398)
for (b in 1:B) {
    z1star = sample(z1, replace = T)
    z2star = sample(z2, replace = T)
    Tstar[b] = abs(t.test(z1star, z2star)$statistic)
}

# Plot the distribution of T* with T.

hist(Tstar, col = 4)
abline(v = T, col = 2, lwd = 2)
box()
Histogram of Tstar
#Compute parametric bootstrap p-value.

```r
mean(Tstar>=T)
[1] 0.2909
```

#Find the critical value of the 0.05 level test.

```r
cv=quantile(Tstar,.95)
cv
2.504745
```

```r
mean(Tstar>=cv)
[1] 0.05
```

#Add the critical value to the plot.

```r
abline(v=cv,col=3,lwd=2)
```
Histogram of Tstar
#Use bootstrap resampling to estimate the power
#of the test when the mean of F1 is 3 units bigger
#than the mean of F2.

pp(z1+3,z2)
B=10000
Tstar=rep(0,B)
set.seed(2448)
for(b in 1:B){
    z1star=sample(z1+3,replace=T)
    z2star=sample(z2,replace=T)
    Tstar[b]=abs(t.test(z1star,z2star)$statistic)
}

mean(Tstar>=cv)
[1] 0.5587

#Use bootstrap resampling to estimate the power
#of the test when the mean of F1 is 5 units bigger
#than the mean of F2.

pp(z1+5,z2)
B=10000
Tstar=rep(0,B)
set.seed(2888)
for(b in 1:B){
    z1star=sample(z1+5,replace=T)
    z2star=sample(z2,replace=T)
    Tstar[b]=abs(t.test(z1star,z2star)$statistic)
}

mean(Tstar>=cv)
[1] 0.9385
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 \( X_1, \ldots, X_n \sim F \) and \( \hat{F}_n \) is the empirical cumulative distribution function, \( \hat{F}_n \) is guaranteed to be close to \( F \) for large \( n \).
More formally, suppose each observation is $k \times 1$. The Strong Law of Large Numbers implies

$$\hat{F}_n(x) \xrightarrow{a.s.} F(x) \text{ as } n \to \infty \quad \forall x \in \mathbb{R}^k.$$

In addition, we have

$$\sup_{x \in \mathbb{R}^k} |\hat{F}_n(x) - F(x)| \xrightarrow{a.s.} 0 \text{ as } n \to \infty.$$
How large should $n$ be in order for $\hat{F}_n$ to be a good estimator of $F$?

That depends on $F$. 
There is much more to learn about the bootstrap.

The books below provide good starting points.
