Resampling Methods

Resampling

The term *resampling* refers to a class of statistical methods that aim to perform inference based on samples drawn from the data set at hand. It is an approach that has only become feasible in the digital computer age and hence has been the subject of a great deal of research in the past 20 years.

Permutation tests are simple examples of resampling techniques.

More generally, resampling approaches can be useful in many problems where the mathematics required for inference is either very difficult or even impossible.

We will discuss three resampling approaches: the Jackknife, the Bootstrap and cross-validation.

Reference:

Efron (1979) "Bootstrap methods: another look at the jackknife", Annals of Statistics, 7:1-26.

Shao and Tu (1995), "The Jackknife and Bootstrap", Springer-Verlag.

Efron and Tibshirani (1993), "An introduction to the bootstrap", Chapman and Hall.

The Jackknife

Quenoulle (1949) (*"Approximation tests of correlation in time series", JRSSB, 11, 18-49*) introduced the jackknife to estimate the bias of an estimator. The basic idea involves recalculating some statistic of interest for reduced data sets obtained by deleting one datum at a time from the original data set. More precisely, let $T_n = T_n(X_1, \cdots, X_n)$ be an estimator of an unknown parameter θ. The bias of T_n is

$$\text{bias}(T_n) = E(T_n) - \theta.$$

Let $T_{(n-1),i} = T_{n-1}(X_1, \cdots, X_{i-1}, X_{i+1}, \cdots, X_n)$ be the estimator of θ based on all of the observations excluding X_i. Then, Quenoulle's jackknife bias estimator is:

$$b_{JACK} = (n-1)(\overline{T}_n - T_n),$$

where $\overline{T}_n = n^{-1} \sum_{i=1}^{n} T_{(n-1),i}$.

This leads to a bias-reduced jackknife estimator,

$$T_{JACK} = T_n - b_{JACK} = nT_n - (n-1)\overline{T}_n.$$

Example 1.1

Consider estimating μ^2 based on the following small sample:

$$1, 2, 7, 16, 19.$$

The sample mean is $\hat{\mu} = 9$, so that the MLE estimate of μ^2 is $\hat{\mu}^2 = 81$.

Let $\hat{u}_{(i)}$ = square of the sample mean with the ith observation omitted, so

$$\begin{aligned}
\hat{\mu}_{(1)} &= \text{square of mean of } 2, 7, 16, 19 = 121, \\
\hat{\mu}_{(2)} &= \text{square of mean of } 1, 7, 16, 19 = 115.56, \\
\hat{\mu}_{(3)} &= \text{square of mean of } 1, 2, 16, 19 = 90.25, \\
\hat{\mu}_{(4)} &= \text{square of mean of } 1, 2, 7, 19 = 52.56, \\
\hat{\mu}_{(5)} &= \text{square of mean of } 1, 2, 7, 16 = 42.25.
\end{aligned}$$

The numbers $\hat{\mu}_{(1)}, \hat{\mu}_{(2)}, \hat{\mu}_{(3)}, \hat{\mu}_{(4)}, \hat{\mu}_{(5)}$ are called the *jackknife replicates of* $\hat{\mu}^2$.

The average of the replicates is 84.32, so that the estimated bias of $\hat{\mu}^2$ is

$4 * (84.32 - 81) = 13.3$.

The bias-reduced jackknife estimator is $5 * 81 - 4 * 84.32 = 67$.

Note that we can achieve the same results with the following `Splus` commands:

```
msq_function(x){mean(x)^2}
jackknife(x,msq)
```

Heuristic Justification of Jackknife:

Suppose

$$\text{bias}(T_n) = \frac{a}{n} + \frac{b}{n^2} + O\left(\frac{1}{n^3}\right),$$

where a and b are unknown, but not involving n.

Then,

$$\text{bias}(\bar{T}_n) = \text{bias}(T_{(n-1),i}) = \frac{a}{n-1} + \frac{b}{(n-1)^2} + O\left(\frac{1}{(n-1)^3}\right).$$

Hence,

$$\begin{aligned} E(b_{JACK}) &= (n-1)\left[\text{bias}(\bar{T}_n) - \text{bias}(T_n)\right] \\ &= (n-1)\left[\left(\frac{1}{n-1} - \frac{1}{n}\right)a + \left(\frac{1}{(n-1)^2} - \frac{1}{n^2}\right)b + O\left(\frac{1}{n^3}\right)\right] \\ &= \frac{a}{n} + \frac{(2n-1)b}{n^2(n-1)} + O\left(\frac{1}{n^2}\right). \end{aligned}$$

The Jackknife became much more popular after Tukey discovered that the bootstrap replicates can be used to estimate the standard error of an estimator (see *"Bias and confidence in not quite large samples", Annals of Mathematical Statistics, 29, 614, 1958*). Writing T_{JACK} as

$$T_{JACK} = \frac{1}{n}\sum_{i=1}^{n}\left[nT_n - (n-1)T_{(n-1),i}\right],$$

Tukey defined *Jackknife pseudovalues* for $i = 1, \cdots, n$:

$$\widetilde{T}_{n,i} = nT_n - (n-1)T_{(n-1),i},$$

and conjectured that:

1. The pseudovalues may be treated as iid.

2. $\widetilde{T}_{n,i}$ has approximately the same variance as $\sqrt{n}T_n$.

It follows that a natural estimator of $\mathrm{Var}(T_n)$ is:

$$\begin{aligned} v_{JACK} &= \frac{1}{n(n-1)} \sum_{i=1}^{n} \left(\widetilde{T}_{n,i} - \frac{1}{n} \sum_{i=1}^{n} \widetilde{T}_{n,i} \right)^2 \\ &= \frac{n-1}{n} \sum_{i=1}^{n} \left(T_{(n-1),i} - \frac{1}{n} \sum_{i=1}^{n} T_{(n-1),i} \right)^2 . \end{aligned}$$

Example 1.2

$T_n = \bar{X}_n$ as an estimator of the population mean, μ:

It is easy to show that $T_{JACK} = \bar{X}_n$, $\widetilde{T}_{n,i} = X_i$ and v_{JACK} corresponds to the traditional estimator.

Example 1.3

$T_n = \bar{X}_n^2$ is an estimator of μ^2:

Some straightforward algebra establishes that $b_{JACK} = S^2/n$.

We can also show that

$$v_{JACK} = \frac{4\bar{X}_n^2\hat{\mu}_2}{n} - \frac{4\bar{X}_n\hat{\mu}_3}{n(n-1)} + \frac{\hat{\mu}_4}{n(n-1)^2} - \frac{\hat{\mu}_2^2}{n^2(n-1)},$$

where μ_k represents the kth central moment.

This has the same first order term as the theoretical variance of $\bar{X}_n^2$:

$$\text{Var}\left(\bar{X}_n^2\right) = \frac{4\mu^2\mu_2}{n} + \frac{4\mu\mu_3}{n^2} + \frac{\mu_4}{n^3}.$$

Example 1.4

The α-trimmed sample mean is given by

$$T_n = \bar{X}_n^{(\alpha)} = \frac{1}{n - 2[n\alpha]} \sum_{i=[n\alpha]+1}^{n-[n\alpha]} X_{(i)},$$

where $[t]$ is the integer part of t and $X_{(i)}$ is the ith order statistic.

When does the Jackknife work?
We present the result here that the jackknife variance estimator is valid for "smooth" functions of the sample mean.

Consistency of Jackknife Variance Estimators

Suppose $X_1, \cdots, X_n$ are iid observations from a p-dimensional distribution, F, with mean $\boldsymbol{\mu}$ and finite covariance matrix $\boldsymbol{\Sigma}$. Suppose that $T_n = \boldsymbol{g}(\bar{X}_n)$, where $\bigtriangledown \boldsymbol{g}$ is defined in the neighborhood of $\boldsymbol{\mu}$, $\bigtriangledown \boldsymbol{g}(\boldsymbol{\mu}) \neq 0$, and $\bigtriangledown \boldsymbol{g}$ is continuous at $\boldsymbol{\mu}$. Then the jackknife variance

estimator for $\bar{T}_n$ is strongly consistent in the sense that

$$\frac{v_{JACK}}{\sigma_n^2} \xrightarrow{a.s.} 1$$

where

$$\sigma_n^2 = n^{-1} \bigtriangledown \boldsymbol{g}(\boldsymbol{\mu})' \boldsymbol{\Sigma} \bigtriangledown \boldsymbol{g}(\boldsymbol{\mu}).$$

Proof: See handout.

Notes:

- Major computational advantage over standard analytical approaches.
- Jackknife fails in some settings. For example,

$$\frac{v_{JACK}}{\sigma^2/n} \xrightarrow{D} \left(\chi^2_2/2\right)^2$$

where v_{JACK} is the jackknife variance of the sample median, and σ^2 is the asymptotic variance of the sample median.

The Bootstrap

Although a data set of size n has $2^n - 1$ non-empty subsets, the jackknife uses only n of them. Hartigan (1969) suggested improvements of the

jackknife that used more of the possible subsets. The landmark paper by Efron in 1979 found a way around the computational problems with Hartigan's approach and revolutionized applied statistics in the process.

Since Efron's paper, the bootstrap has had many variations developed, including the *double* bootstrap and so on. However, each are variations on the same theme of a general resampling technique that can be applied to almost any estimation procedure as a means of making inferential statements. For example, the procedure works for the sample median. The name "bootstrap" comes from the saying: "when all else fails, pull yourself up by your bootstraps".

We will explain the basic idea of the bootstrap through an example. Consider the dataset of 5 "failure times":

$$1, 2, 7, 16, 19.$$

These data were, in fact, generated from the exponential distribution with scale parameter $\frac{1}{10}$, so in this case, we know that the underlying

probability structure is given by

$$F(x) = 1 - e^{-\frac{x}{10}}.$$

Suppose that we wish to make inference about the median failure time for this distribution based on our data. The observed sample median is, of course,

$$\widehat{M} = 7.$$

But how about a measure of variability, such as a standard error? This would require calculation of

$$\text{Var}(\widehat{M}).$$

It can be shown showed that

$$\sqrt{n}(\widehat{M} - M) \xrightarrow{D} N\left(0, \frac{1}{4f(M)^2}\right),$$

or, equivalently,

$$\frac{\widehat{M} - M}{\frac{1}{2f(M)\sqrt{n}}} \xrightarrow{D} N(0,1).$$

It is relatively straightforward to extend this proof to handle any *sample quantile*,

not just the sample median. A common notation for the pth population quantile is

$$\xi_p = F^{-1}(p), \quad 0 < p < 1.$$

For example $\xi_{1/2}$ is the median and $\xi_{1/4}$ and $\xi_{3/4}$ are quartiles. The obvious estimator for ξ_p is

$$\hat{\xi}_p = F_n^{-1}(p),$$

where $F_n(x)$ is the empirical cdf. We note that there are several variants of this definition used in the textbooks and packages.

Asymptotic Distribution of a Sample Quantile

Let $0 < p < 1$. If F posses a density in a neighborhood of ξ_p and if f is positive and continuous at ξ_p, then

$$\sqrt{n}(\hat{\xi}_p - \xi_p) \xrightarrow{D} N\left(0, \frac{p(1-p)}{f(\xi_p)^2}\right).$$

So one possible estimate of $\mathrm{Var}(\widehat{M})$ is

$$\frac{1}{4f(\widehat{M})^2 \times 5} = 20.276,$$

which uses the fact that the data have density

$$f(x) = \frac{1}{10} e^{-x/10}.$$

Apart from the problem that we are using our artificial knowledge of the underlying density function, this estimator of variance suffers from the fact that the sample size is probably not large enough for the Central Limit Theorem approximation to be very accurate.

Since we are working under the assumption that the underlying data are from an exponential distribution, we could probably work out the exact distribution of the sample median. However, this would involve some difficult numerical integration. Since we are pretending that we know the underlying distribution function we could get around this by simulation as follows:

- Simulate A new samples from the Exponential$(1/10)$ distribution where A is a very large number (e.g., 10000).
- For each sample, $a = 1, \cdots, A$, compute the sample median $\widehat{M}_a$.
- Estimate $\text{Var}(\widehat{M})$ using

$$\widehat{\text{Var}}(\widehat{M})_A = \frac{1}{A-1} \sum_{a=1}^{A} \left\{ \widehat{M}_a - \overline{\widehat{M}} \right\}^2.$$

It follows from the Weak Law of Large Numbers that

$$\widehat{\text{Var}}(\widehat{M})_A \xrightarrow{P} \text{Var}(\widehat{M}) \quad \text{as} \quad A \to \infty,$$

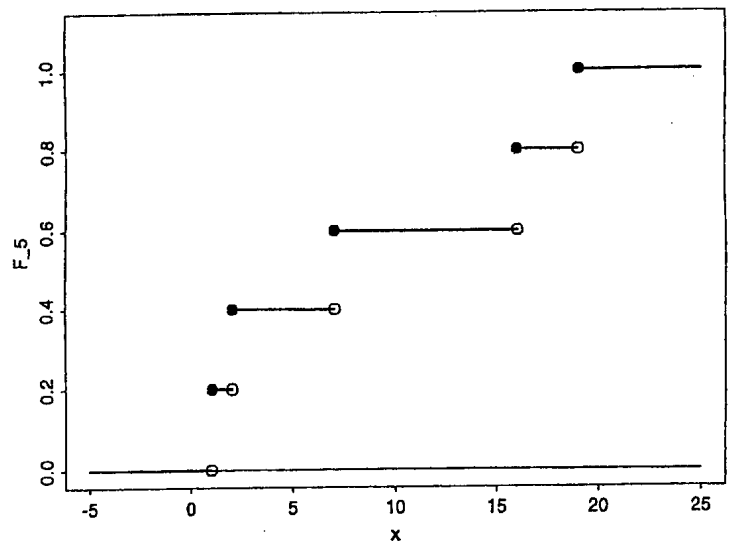
so provided we take A large enough, we should be able to get (in a probabilistic sense) as close as we like to $\text{Var}(\widehat{M})$.

The only problem is that, in practice, we don't know F. However, we do have a good estimator for F, namely the empirical distribution function F_n.

The *essential idea of the bootstrap is that we sample from* F_n *instead of* F.

Because of the Glivenko-Cantelli Theorem, we hope that the results using F_n are not too far from those using F.

For our example dataset, the empirical distribution function F_5 is shown in the following figure:



This is actually the distribution function of a discrete random variable D which equals 1, 2, 7, 16, or 19 with equal probabilities of $\frac{1}{5}$.

$$f_D(d) = \begin{cases} \frac{1}{5} & d = 1 \\ \frac{1}{5} & d = 2 \\ \frac{1}{5} & d = 7 \\ \frac{1}{5} & d = 16 \\ \frac{1}{5} & d = 19. \end{cases}$$

Drawing a sample of size 5 from this distribution is equivalent to drawing a sample *with replacement* from the set

$$\{1, 2, 7, 16, 19\}.$$

So we could repeat the algorithm given above to estimate $\text{Var}(\widehat{M})$. Actually, for this example, we don't really need to do 10,000 replications since there are only 126 unique samples of F_5. These are listed in the following table, along with their probabilities. Suppose that we define the random vector $\boldsymbol{x}^*$ to assume each one of these table values with

probabilities as listed. The distribution of $\boldsymbol{x}^*$ is named the *bootstrap distribution* of the original sample.

Table 1: Bootstrap distribution of the sample 1, 2, 7, 16, 19.

1	1	1	1	1	(1/3125)	1	2	2	16	16	(30/3125)	2	2	7	16	19	(60/3125)
1	1	1	1	2	(5/3125)	1	2	2	16	19	(60/3125)	2	2	7	19	19	(30/3125)
1	1	1	1	7	(5/3125)	1	2	2	19	19	(30/3125)	2	2	16	16	16	(10/3125)
1	1	1	1	16	(5/3125)	1	2	7	7	7	(20/3125)	2	2	16	16	19	(30/3125)
1	1	1	1	19	(5/3125)	1	2	7	7	16	(60/3125)	2	2	16	19	19	(30/3125)
1	1	1	2	2	(10/3125)	1	2	7	7	19	(60/3125)	2	2	19	19	19	(10/3125)
1	1	1	2	7	(20/3125)	1	2	7	16	16	(60/3125)	2	7	7	7	7	(5/3125)
1	1	1	2	16	(20/3125)	1	2	7	16	19	(120/3125)	2	7	7	7	16	(20/3125)
1	1	1	2	19	(20/3125)	1	2	7	19	19	(60/3125)	2	7	7	7	19	(20/3125)
1	1	1	7	7	(10/3125)	1	2	16	16	16	(20/3125)	2	7	7	16	16	(30/3125)
1	1	1	7	16	(20/3125)	1	2	16	16	19	(60/3125)	2	7	7	16	19	(60/3125)
1	1	1	7	19	(20/3125)	1	2	16	19	19	(60/3125)	2	7	7	19	19	(30/3125)
1	1	1	16	16	(10/3125)	1	2	19	19	19	(20/3125)	2	7	16	16	16	(20/3125)

Table 1 (continued): Bootstrap distribution of the sample $1,\ 2,\ 7,\ 16,\ 19$.

1	1	1	16	19	(20/3125)	1	7	7	7	7	(5/3125)	2	7	16	16	19	(60/3125)
1	1	1	19	19	(10/3125)	1	7	7	7	16	(20/3125)	2	7	16	19	19	(60/3125)
1	1	2	2	2	(10/3125)	1	7	7	7	19	(20/3125)	2	7	19	19	19	(20/3125)
1	1	2	2	7	(30/3125)	1	7	7	16	16	(30/3125)	2	16	16	16	16	(5/3125)
1	1	2	2	16	(30/3125)	1	7	7	16	19	(60/3125)	2	16	16	16	19	(20/3125)
1	1	2	2	19	(30/3125)	1	7	7	19	19	(30/3125)	2	16	16	19	19	(30/3125)
1	1	2	7	7	(30/3125)	1	7	16	16	16	(20/3125)	2	16	19	19	19	(20/3125)
1	1	2	7	16	(60/3125)	1	7	16	16	19	(60/3125)	2	19	19	19	19	(5/3125)
1	1	2	7	19	(60/3125)	1	7	16	19	19	(60/3125)	7	7	7	7	7	(1/3125)
1	1	2	16	16	(30/3125)	1	7	19	19	19	(20/3125)	7	7	7	7	16	(5/3125)
1	1	2	16	19	(60/3125)	1	16	16	16	16	(5/3125)	7	7	7	7	19	(5/3125)
1	1	2	19	19	(30/3125)	1	16	16	16	19	(20/3125)	7	7	7	16	16	(10/3125)
1	1	7	7	7	(10/3125)	1	16	16	19	19	(30/3125)	7	7	7	16	19	(20/3125)
1	1	7	7	16	(30/3125)	1	16	19	19	19	(20/3125)	7	7	7	19	19	(10/3125)
1	1	7	7	19	(30/3125)	1	19	19	19	19	(5/3125)	7	7	16	16	16	(10/3125)
1	1	7	16	16	(30/3125)	2	2	2	2	2	(1/3125)	7	7	16	16	19	(30/3125)
1	1	7	16	19	(60/3125)	2	2	2	2	7	(5/3125)	7	7	16	19	19	(30/3125)
1	1	7	19	19	(30/3125)	2	2	2	2	16	(5/3125)	7	7	19	19	19	(10/3125)
1	1	16	16	16	(10/3125)	2	2	2	2	19	(5/3125)	7	16	16	16	16	(5/3125)
1	1	16	16	19	(30/3125)	2	2	2	7	7	(10/3125)	7	16	16	16	19	(20/3125)
1	1	16	19	19	(30/3125)	2	2	2	7	16	(20/3125)	7	16	16	19	19	(30/3125)
1	1	19	19	19	(10/3125)	2	2	2	7	19	(20/3125)	7	16	19	19	19	(20/3125)
1	2	2	2	2	(5/3125)	2	2	2	16	16	(10/3125)	7	19	19	19	19	(5/3125)
1	2	2	2	7	(20/3125)	2	2	2	16	19	(20/3125)	16	16	16	16	16	(1/3125)
1	2	2	2	16	(20/3125)	2	2	2	19	19	(10/3125)	16	16	16	16	19	(5/3125)
1	2	2	2	19	(20/3125)	2	2	7	7	7	(10/3125)	16	16	16	19	19	(10/3125)
1	2	2	7	7	(30/3125)	2	2	7	7	16	(30/3125)	16	16	19	19	19	(10/3125)
1	2	2	7	16	(60/3125)	2	2	7	7	19	(30/3125)	16	19	19	19	19	(5/3125)
1	2	2	7	19	(60/3125)	2	2	7	16	16	(30/3125)	19	19	19	19	19	(1/3125)

The next table shows the distribution of the medians obtained from each member of Table 1. This is called the *bootstrap distribution* of the sample median.

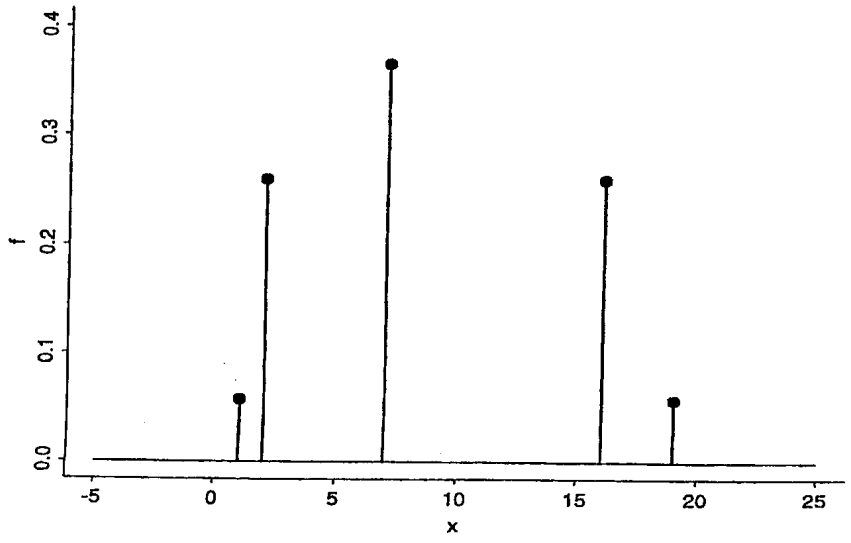
Table 2: Bootstrap distribution of the sample median based on the sample $1,\ 2,\ 7,\ 16,\ 19$ obtained from taking the median of each sample in Table 1.

1	(1/3125)	2	(60/3125)	2	(30/3125)	16	(60/3125)	7	(60/3125)	7	(1/3125)
1	(5/3125)	2	(30/3125)	2	(60/3125)	19	(20/3125)	7	(30/3125)	7	(5/3125)
1	(5/3125)	2	(60/3125)	2	(30/3125)	16	(5/3125)	16	(10/3125)	7	(5/3125)
1	(5/3125)	2	(30/3125)	7	(20/3125)	16	(20/3125)	16	(30/3125)	7	(10/3125)
1	(5/3125)	7	(10/3125)	7	(60/3125)	16	(30/3125)	16	(30/3125)	7	(20/3125)
1	(10/3125)	7	(30/3125)	7	(60/3125)	19	(20/3125)	19	(10/3125)	7	(10/3125)
1	(20/3125)	7	(30/3125)	7	(60/3125)	19	(5/3125)	7	(5/3125)	16	(10/3125)
1	(20/3125)	7	(30/3125)	7	(120/3125)	2	(1/3125)	7	(20/3125)	16	(30/3125)
1	(20/3125)	7	(60/3125)	7	(60/3125)	2	(5/3125)	7	(20/3125)	16	(30/3125)
1	(10/3125)	7	(30/3125)	16	(20/3125)	2	(5/3125)	7	(30/3125)	19	(10/3125)
1	(20/3125)	16	(10/3125)	16	(60/3125)	2	(5/3125)	7	(60/3125)	16	(5/3125)
1	(20/3125)	16	(30/3125)	16	(60/3125)	2	(10/3125)	7	(30/3125)	16	(20/3125)
1	(10/3125)	16	(30/3125)	19	(20/3125)	2	(20/3125)	16	(20/3125)	16	(30/3125)
1	(20/3125)	19	(10/3125)	7	(5/3125)	2	(20/3125)	16	(60/3125)	19	(20/3125)
1	(10/3125)	2	(5/3125)	7	(20/3125)	2	(10/3125)	16	(60/3125)	19	(5/3125)
2	(10/3125)	2	(20/3125)	7	(20/3125)	2	(20/3125)	19	(20/3125)	16	(1/3125)
2	(30/3125)	2	(20/3125)	7	(30/3125)	2	(10/3125)	16	(5/3125)	16	(5/3125)
2	(30/3125)	2	(20/3125)	7	(60/3125)	7	(10/3125)	16	(20/3125)	16	(10/3125)
2	(30/3125)	2	(30/3125)	7	(30/3125)	7	(30/3125)	16	(30/3125)	19	(10/3125)
2	(30/3125)	2	(60/3125)	16	(20/3125)	7	(30/3125)	19	(20/3125)	19	(5/3125)
2	(60/3125)	2	(60/3125)	16	(60/3125)	7	(30/3125)	19	(5/3125)	19	(1/3125)

From this table, it follows that the bootstrap distribution of the median has probability function

$$f(x) = \begin{cases} \frac{181}{3125} & x = 1 \\ \frac{811}{3125} & x = 2 \\ \frac{1141}{3125} & x = 7 \\ \frac{811}{3125} & x = 16 \\ \frac{181}{3125} & x = 19, \end{cases}$$

and is given in the following.


f
0.0
0.1
0.2
0.3
0.4
-5
0
5
10
15
20
25
x

It is straightforward to work out the population variance corresponding to the bootstrap distribution of the sample median. It is

$$36.01483.$$

This is the *exact bootstrap estimator* of $\mathrm{Var}(\widehat{M})$ and can be written as

$$\widehat{\mathrm{Var}(\widehat{M})}_{boot} = 36.01483.$$

The reason why we could work out the exact bootstrap distribution of $\widehat{M}$ is that the sample size is so small, and even then, it took a lot of work. For a sample of size n, it can be shown that the bootstrap distribution has

$$\binom{2n-1}{n-1}$$

distinct values (Hall, 1987). The next table shows this value for selected values of n.

Table 3: # distinct values in the bootstrap distribution from a sample of size n.

n	$\binom{2n-1}{n-1}$
5	126
10	92,378
20	68,923,264,410
50	5.045×10^{28}
100	4.527×10^{58}

We see from this that it is infeasible for most practical sample sizes to compute the exact bootstrap estimator of variance. Instead we simulate a large number of samples from it. This can be done as follows:

1. Draw B samples with replacement from 1, 2, 7, 16, 19.
2. For each sample $b = 1, \cdots, B$, compute the sample median $\widehat{M}_b$.
3. The bootstrap estimate of variance is

$$\widehat{\mathrm{Var}(\widehat{M})}_B = \frac{1}{B-1} \sum_{b=1}^{B} \left\{ \widehat{M}_b - \overline{\widehat{M}} \right\}^2,$$

where

$$\overline{\widehat{M}} = \frac{1}{B}\sum_{b=1}^{B}\widehat{M}_b.$$

Example 1.5

Consider the pressure vessel data:

Table 4: Failure Times for Fibre/Epoxy Pressure Vessels (hours)

274	28.5	1.7	20.8
871	363	1311	1661
236	828	458	290
54.9	175	1787	970
.75	1278	776	126

The sample median is 326.5.

The following `S-PLUS` code obtains a bootstrap variance estimator for $\widehat{M}$:

```
##########################################
 # S-PLUS code for computing a bootstrap
 # estimator of the variance of the median
 # for the pressure vessel data. The data
 #are stored in an array named "pressure".

 pressure_c(274,28.5,1.7,20.8,871,363,1311 ,1661 ,236,
 828,458,290,54.9,175,1787,970,.75,1278,776 ,126)

 n <- length(pressure)

 seed <- 2909 set.seed(seed) nrep <- 1000

 median.boot <- numeric()

 for (b in 1:nrep)
 {
   pressure.boot <- sample(pressure,n,replace=T)
   median.boot[b] <- median(pressure.boot)
 }

 var.boot <- var(median.boot)

 OR

 bootstrap(pressure, median)
##########################################
```

The answer is

$$\widehat{\text{Var}(\widehat{M})}_{boot} = 35864.8312.$$

Therefore, we can state an approximate 95% confidence interval for the population median as:

$$326.5 \pm 1.96 \times \sqrt{35864.8312} = (-44.69, 697.69).$$

General Result

Let $\boldsymbol{X} = (X_1, \cdots, X_n)$ be an iid sample and suppose we are interested in

$$\hat{\theta} = s(\boldsymbol{X})$$

for some functions $s : \Re^n \to \Re$. We can write the bootstrap algorithm for estimation of standard errors as:

The Bootstrap Estimate of Standard Error

1. Select B independent bootstrap samples

$$\boldsymbol{X}_1^*, \cdots, \boldsymbol{X}_B^*.$$

2. Obtain the estimate of θ for each bootstrap sample:

$$\hat{\theta}_b^* = s(\boldsymbol{X}_b^*), \quad b = 1, \cdots, B.$$

   These are called the *bootstrap replications* of $\hat{\theta}$.

3. Estimate the standard error of $\hat{\theta}$ by using the sample standard deviation of the B bootstrap replications.

$$\widehat{se(\hat{\theta})} = \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}\left\{\hat{\theta}_b^* - \overline{\hat{\theta}}^*\right\}^2},$$

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

Note that the same principle applies just as easily to multivariate

datasets as we will now see through a number of examples.

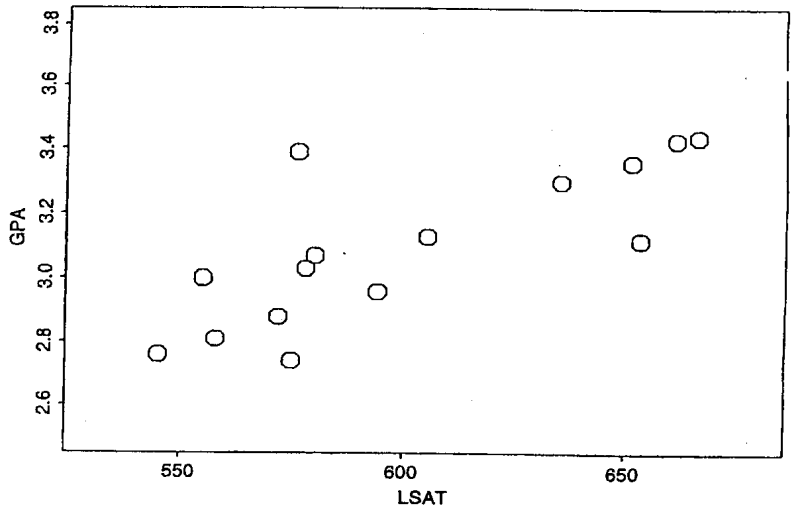
Example 1.6

The following table shows data on two different scores for 15 law schools, taken from Efron and Tibshirani. The table contains 15 pairs of LSAT scores (the average score for the class on a nation law test) and GPA (average undergraduate grade-point average for the class)

Table 5: The law school data

LSAT	GPA
576	3.39
635	3.30
558	2.81
578	3.03
666	3.44
580	3.07
555	3.00
661	3.43
651	3.36
605	3.13
653	3.12
575	2.74
545	2.76
572	2.88
594	2.96

The data are plotted in the following figure:


GPA
3.8
3.6
3.4
3.2
3.0
2.8
2.6
550
600
650
LSAT

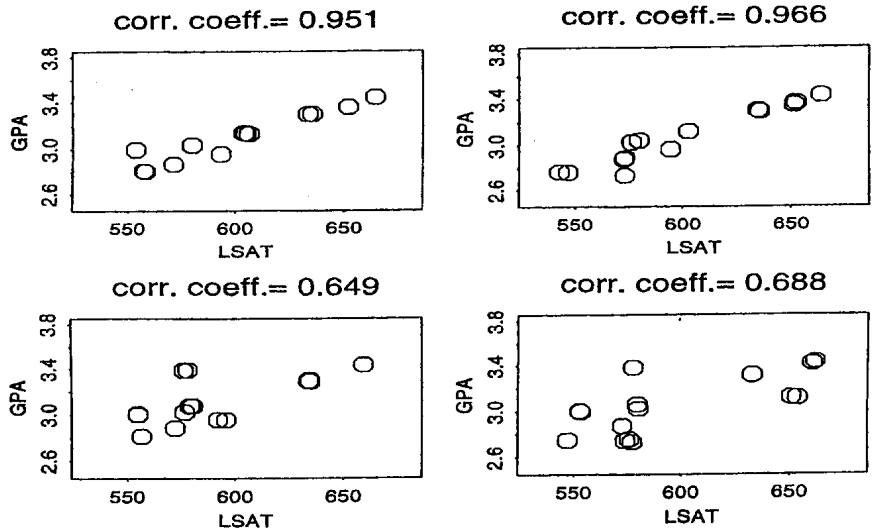
The main interest in this example centers on the correlation coefficient between LSAT and GPA. The point estimator is

$$\hat{\rho} = .776.$$

We will now appeal to the bootstrap to give us a standard error estimate without having to formulate a model, or perform any fancy mathematics.

The easiest way to understand bootstrap resampling for multivariate data is to think of each multivariate observation as being a "location in space". Univariate observations live on a line, bivariate observations live on a plane, trivariate samples live in three-dimensional space, etc... A bootstrap resample simply corresponds to a sample of those locations.

This idea is illustrated in the next figure, where 4 bootstrap resamples are plotted (with some jittering to distinguish points at the same location). The corresponding sample correlation coefficients are also shown.


corr. coeff.= 0.951
corr. coeff.= 0.966
corr. coeff.= 0.649
corr. coeff.= 0.688
GPA
LSAT
2.6
3.0
3.4
3.8
550
600
650

When it comes to programming, the easiest approach is to store the data in a matrix $\boldsymbol{X}$ where each column corresponds to each variable. For the law school data, the data matrix is (with row numbers labelled):

$$\boldsymbol{X} = \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \\ 12 \\ 13 \\ 14 \\ 15 \end{array} \begin{bmatrix} 576 & 3.39 \\ 635 & 3.30 \\ 558 & 2.81 \\ 578 & 3.03 \\ 666 & 3.44 \\ 580 & 3.07 \\ 555 & 3.00 \\ 661 & 3.43 \\ 651 & 3.36 \\ 605 & 3.13 \\ 653 & 3.12 \\ 575 & 2.74 \\ 545 & 2.76 \\ 572 & 2.88 \\ 594 & 2.96 \end{bmatrix}$$

Since bootstrap resampling of a multivariate dataset involves moving points around in space, we should really perform the resampling on the row indices:

$$\{1,\ 2,\ 3, \ldots,\ 15\}$$

rather than on the data themselves. In S−PLUS, we can do this through the command:

```
inds.boot <-sample(1:15,15,replace=T)
```

For example, we might get

```
8 15 5 12 14 5 7 6 14 6 11 14 6 12 14
```

This would result in the bootstrap sample

$$\boldsymbol{X}^* = \begin{array}{c} 8 \\ 15 \\ 5 \\ 12 \\ 14 \\ 5 \\ 7 \\ 6 \\ 14 \\ 6 \\ 11 \\ 14 \\ 6 \\ 12 \\ 14 \end{array} \begin{bmatrix} 661 & 3.43 \\ 594 & 2.96 \\ 666 & 3.44 \\ 575 & 2.74 \\ 572 & 2.88 \\ 666 & 3.44 \\ 555 & 3.00 \\ 580 & 3.07 \\ 572 & 2.88 \\ 580 & 3.07 \\ 653 & 3.12 \\ 572 & 2.88 \\ 580 & 3.07 \\ 575 & 2.74 \\ 572 & 2.88 \end{bmatrix}$$

This maneuver can be done in S−PLUS as follows:

```
X.boot <- X[inds.boot,]
```

The following S−PLUS function `boot.se()` computes a bootstrap

standard error estimate for a general estimator based on a multivariate dataset stored in a matrix `x`:

```
########## S function: boot.se ##########

# For obtaining a bootstrap estimate of # the standard error of an
estimator, # defined by the function "estimator" # which takes, as
argument, a matrix "x" # and returns a scalar value.

# Last changed: 7th April, 1998

boot.se <-function(x,estimator,num.rep = 200) {
   # Convert x to a matrix in case it is an array.

   x <- as.matrix(x)

   # Set preliminary quantities

   n <- nrow(x)
   estimator.boot <- numeric()

   # Obtain bootstrap replications of estimator.
```

```
   for(b in 1:num.rep)
   {
      inds.boot <- sample(1:n,n,replace=T)  # sample from {1,..,n} with r'ment.
      x.boot <- x[inds.boot,]               # obtain bootstrap sample
      estimator.boot[b] <- estimator(x.boot)# obtain bootstrap estimator
   }

   # Return the sample standard deviation of the
   # bootstrap replications.

   return(sqrt(var(estimator.boot)))
}

######### End of boot.se ##########
```

One of the nice things about S−PLUS is that we can pass an S−PLUS *function* through the argument `estimator`. Therefore, the standard error for the pressure vessel example can be found through the call:

```
boot.se(pressure,median)
```

since `median()` is an in-built function that operates on an $n \times 1$ matrix (or an `S-PLUS` array).

Let's now return to the law school data and the problem of estimating the standard error of the correlation coefficient. `S-PLUS` has a function named `cor()` which returns the sample correlation coefficient of data stored in arrays `x1` and `x2` through the call:

```
cor(x1,x2)
```

This is not quite in the right format for `boot.se()`, so we will need to define a new function that takes as its argument an $n \times 2$ matrix and returns the sample correlation coefficient of the two columns:

```
corr.coeff <- function(x) {
   return(cor(x[,1],x[,2]))
}
```

Now that we've done this we can find the point estimate and standard error through the code:

```
point.est <- corr.coeff(law.data) se.est <-
boot.se(law.data,corr.coeff)
```

This yielded:

$$\widehat{\text{se}(\hat{\rho})}_{\text{boot}} = .141,$$

giving a standard confidence interval of

$$(.500, 1.05).$$

There is most definitely a positive correlation between `LSAT` and `GPA` for law schools.

Example 1.7

The following table shows Birth weights (kg) and occurrence of bronchopulmonary dysplasia (1=occurred, 0=not occurred) for fourteen low birth weight infants. We are interested in applying the logistic regression model:

$$P(\text{BPD} = 1|\text{birthweight}) = \frac{e^{\beta_0+\beta_1 \text{birthweight}}}{1 + e^{\beta_0+\beta_1 \text{birthweight}}},$$

so that the main parameter of interest is the slope coefficient β_1, which quantifies the association between bronchopulmonary dysplasia and birthweight.

birth weight (kg)	BPD
1.06	1
1.62	0
1.47	0
.84	1
.76	1
.91	1
1.11	0
1.26	0
1.34	0
1.46	0
.85	1
1.53	0
1.00	0
1.54	0

To obtain a bootstrap estimate of the standard error of $\hat{\beta}_1$ using `boot.se()`, we store the regression data in an $n \times 2$ matrix `BPD.data` and define the function:

```
slope.est.logist <- function(x) {
   fit <- glm(x[,2]~x[,1],family="binomial")

   return(fit$coef[2])
}
```

Then all we need to type (in theory) is:

```
point.est <- slope.est.logist(BPD.data)

se.est <- boot.se(BPD.data,slope.est.logist)
```

This leads to

$$\hat{\beta}_1 = -19.67 \qquad \text{and} \qquad \widehat{\text{se}(\hat{\beta}_1)}_{\text{boot}} = 80.722.$$

However, be aware that the bootstrap approach can run into difficulties in the logistic regression setting since some of the bootstrap replications lead to numerically unstable fits.

The strength of the Bootstrap lies in complex problems where normal inference would be hard. For example, Efron and Tibshirani give an example of a multivariate analysis of the *score data*. These data consist of scores that $n = 88$ students obtained on each of 5 examinations, in mechanics, vectors, algebra, analysis, and statistics.

To use `boot.se()` to obtain a standard error estimate for the estimate of the largest relative eigenvalue, we need the `S-PLUS` function:

```
largest.relative.eigenvalue <- function(x)
{
   eigenvalues <- eigen(var(x))$values
   return(max(eigenvalues)/sum(eigenvalues))
}
```

Assuming that the score data are stored in an 88×5 matrix named `testscore.data`, then all we need are the calls:

```
point.est <- largest.relative.eigenvalue(testscore.data)
se.est <-boot.se(testscore.data,largest.relative.eigenvalue)
```

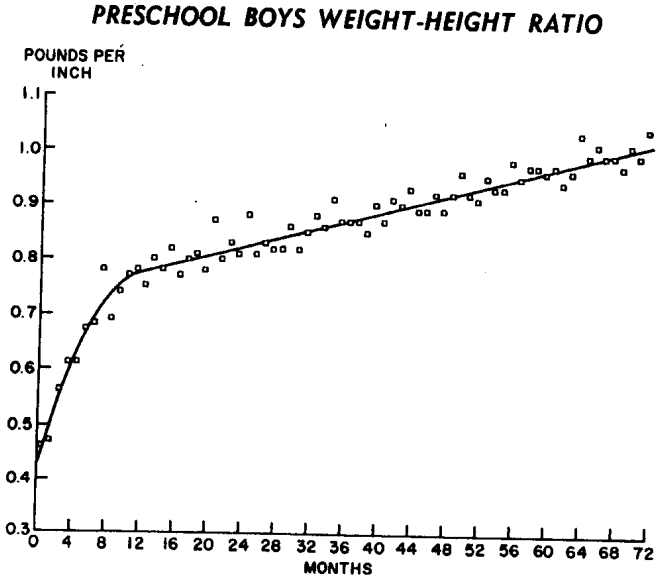
This leads to

$$\hat{\theta} = 0.619 \qquad \text{and} \qquad \widehat{\text{se}(\hat{\theta})}_{\text{boot}} = 0.049,$$

which suggests that the largest relative eigenvalue is in the approximate range 0.5 to 0.7. To test the theory that a single measure (corresponding to the first eigenvector) explains most of the variability, the next part of the analysis should perform inference on the other eigenvalues. However, this requires a multiparameter analysis which we will defer till later.

Example 1.8

The next figure shows a scatterplot of weight-height ratio versus age of 200 preschool boys. The data are extracted from a paper of Gallant and Fuller (1973) (*Journal of the American Statistical Association*).


PRESCHOOL BOYS WEIGHT-HEIGHT RATIO
POUNDS PER INCH
1.1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0 4 8 12 16 20 24 28 32 36 40 44 48 52 56 60 64 68 72
MONTHS

The trend in the figure shows a pronounced "change-point" at about 12 months of age, where the rate of change of the weight-height ratio suddenly becomes smaller. An interesting statistical problem is the estimation of this change-point location. One way to do this is through a "broken stick" regression model:

$$\texttt{ratio} = \beta_0 + \beta_1\,\texttt{age} + \beta_2\,(\texttt{age} - \kappa)_+ + \texttt{error}.$$

The notation u_+ means $\max(u, 0)$ so $(\texttt{age} - \kappa)_+$ represents a function that is zero for $\texttt{age} < \kappa$ and then becomes $\texttt{age} - \kappa$ for $\texttt{age} \geq \kappa$. With a bit of thought, we can see that the above model represents the family of two line segments with a join point, or `knot` at κ – hence the name "broken stick".

The presence of κ in the model leads to a very difficult estimation problem. The paper by Gallant and Fuller gives a complicated algorithm for handling this. Instead, we tried non-linear least squares and obtained

the fit given in the figure. Note that the estimated change-point is:

$$\hat{\kappa} = 12.1722.$$

The non-linear least squares approach has given a reasonable fit, and an estimate of the change-point. But what about inference? It seems that there is no asymptotic theory for $\hat{\kappa}$ since the estimating equations are so "non-regular". There is little choice in this case but to use a resampling procedure. Once again we can use `boot.se()` with the `S-PLUS` function:

```
change.point <- function(x)
{
   y <- x[,2]
   x <- x[,1]

   dat.mat <- as.data.frame(cbind(x,y))

   fit <- nls(y ~ cbind(1, x, pmax(x-knot, 0)),dat.mat,
                  algorithm="plinear",start = c(knot = 12),trace=T)

   return(fit$parameters[1])
}
```

With 48 bootstrap replications we obtained:

$$\widehat{\text{se}(\hat{\kappa})}_{\text{boot}} = 0.213.$$

So the standard confidence interval for the change-point is

$$(11.75, 12.39) \quad \text{months.}$$

The reason for doing only 48 replications is that the non-linear least squares procedure is prone to crashing, and in this case it did on the 49th replication. Like the logistic regression problem, the bootstrap standard error estimation algorithm is prone to numerical problems for complicated estimators. This would require some careful manipulation to avoid.

Bootstrap Pictures

As we argued at the start of the previous section, the standard error is a rather crude summary of the distribution of the estimator $\hat{\theta}$. It is not too bad when the distribution of $\hat{\theta}$ is approximately normally distributed since, in that case the distribution is completely determined by location (estimated by $\hat{\theta}$) and scale (estimated by the standard error of $\hat{\theta}$). But when the distribution is not normal, for example, when it is skewed, then the standard error does not tell the full story. A very appealing feature about the bootstrap is that it can estimate almost any distributional feature of $\hat{\theta}$. In particular, it can give us an estimate of the density function itself by applying a *density estimation technique* to the bootstrap replicates. We call this a *bootstrap picture*. The simplest density estimator is the histogram.

The bootstrap pictures reveal a good deal more than the standard error estimates. For example, in the first two examples, the bootstrap distributions of the estimates are rather skewed. For the third example,

corresponding to the logistic regression, we see that there is a strange clustering effect. This indicates that the estimate of the slope for this logistic regression problem is subject to a high, non-standard, degree of variability.

It is also useful to plot vertical lines corresponding to the actual estimator (solid line) and the sample mean of the bootstrap distribution (dashed line). The latter is found using:

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

These lines are useful since they give an indication of the *bias* of the estimator. We will discuss how the bootstrap can be used to estimate bias in the next section.

The histogram now has a strong competitor, the kernel density estimator, which has better large sample properties. Therefore, we could

also draw bootstrap pictures based on kernel density estimation of the bootstrap replicates.

These kernel density estimation-based bootstrap pictures show the sampling distributions in a way that better reflects the fact that the true sampling distributions have smooth density functions.

Bootstrap pictures require a reliable rule for choosing the amount of "smoothing" of the bootstrap replicates. For the histogram, this amounts to choosing the width of the bins. For kernel density estimation based bootstrap pictures we use the rule of Sheather and Jones (*Journal of the Royal Statistical Society, Series B* (1991)).

Bias Estimation

Like the Jackknife, the Bootstrap can be used to provide an estimate of bias. One way to see this is through the bootstrap picture.

Formally,

$$\text{bias}(\hat{\theta}) = E_F(\hat{\theta}) - \theta.$$

The bootstrap estimate of bias uses the fact that F_n estimates F:

$$\widehat{\text{bias}(\hat{\theta})}_{\text{boot}} = E_{F_n}(\hat{\theta}^*) - \hat{\theta}.$$

Notice that the bootstrap estimator replaces F by F_n, and $\{X_1, \ldots, X_n\}$ by $\{X_1^*, \ldots, X_n^*\}$: *population* information is replaced by *sample* information; *sample* information is replaced by *resample* information.

As for standard errors, except when the sample size is small, it is not feasible to compute $E_{F_n}(\hat{\theta}^*)$ exactly. Instead it is usually estimated by

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

If it appears that the bias is substantial, then an obvious remedy is to correct $\hat{\theta}$ so that its bootstrap bias estimate is zero. This results in

$$\hat{\theta}_{\text{boot,BC}} = \hat{\theta} - \widehat{\text{bias}(\hat{\theta})}_{\text{boot}}.$$

Here "BC" is a commonly used abbreviation for "bias corrected".

As an example, recall the logistic regression involving the BPD data. There, the estimated slope was

$$\hat{\beta}_1 = -19.678.$$

However, recall that the bootstrap picture was that given previously, which indicated that the mean of the $\hat{\theta}$ distribution is to the left of $\hat{\beta}_1$. It is estimated to be

$$\overline{\hat{\theta}}^* = -259.9349.$$

So there is an estimated bias of about

$$\widehat{\text{bias}(\hat{\theta})}_{\text{boot}} = -259.93 - (-19.678) = -240.25$$

slope units. The bootstrap bias corrected estimate is

$$\hat{\theta}_{\text{boot,BC}} = -19.678 - (-240) = 220.322.$$

At first, this might seem counterintuitive, the estimate seems to have gone the "wrong way". But note that we are trying to come up with an estimate that is more likely to equal the mean of its sampling distribution. If we obtained the bootstrap picture for $\hat{\theta}_{\text{boot,BC}}$ then it should be centered about $\hat{\theta}_{\text{boot,BC}}$.

Note, however, that bias correction has its problems. This is essentially due to the fact that biases are harder to estimate than standard errors. (This is explained more fully on page 138 of Efron and Tibshirani (1993).)

The Parametric Bootstrap

The bootstrap strategy that we have dealt with so far essentially uses the idea:

Replace F by F_n and use samples from F_n to make inference about θ.

This method often goes by the fuller title *nonparametric bootstrap* because the true distribution F is replaced by a *nonparametric* estimator F_n.
However, it is often the case that a reasonable parametric model exists for F and this information can be incorporated into the bootstrap idea.

Recall the law school example. It may be reasonable to assume that the data really are bivariate normal, i.e.,

$$\begin{bmatrix} \text{LSAT}_i \\ \text{GPA}_i \end{bmatrix} \sim N_2\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}\right),$$

for some mean vector $\boldsymbol{\mu}$ and some covariance matrix $\boldsymbol{\Sigma}$. Replacing $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ by their usual estimates we get

$$\begin{bmatrix} \text{LSAT}_i \\ \text{GPA}_i \end{bmatrix} \sim N_2\left(\begin{bmatrix} 600.266 \\ 3.095 \end{bmatrix}, \begin{bmatrix} 1746.78 & 7.902 \\ 7.902 & .0593 \end{bmatrix}\right).$$

The parametric bootstrap approach to estimation of $\text{se}(\hat{\rho})$ for this example is:

1. Select B independent bootstrap samples

$$\boldsymbol{X}_1^*, \ldots, \boldsymbol{X}_B^*$$

from the

$$N_2\left(\begin{bmatrix} 600.266 \\ 3.095 \end{bmatrix}, \begin{bmatrix} 1746.78 & 7.902 \\ 7.902 & .0593 \end{bmatrix}\right)$$

distribution.

2. Obtain the estimate of θ for each bootstrap sample:

$$\hat{\theta}_b^* = s(\boldsymbol{X}_b^*), \quad b = 1, \ldots, B.$$

3. Estimate the standard error of $\hat{\theta}$ by using the sample standard deviation of the B bootstrap replications:

$$\widehat{\text{se}(\hat{\theta})} = \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}\left\{\hat{\theta}_b^* - \overline{\hat{\theta}}^*\right\}^2}$$

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

Comparison of this with the algorithm given before shows that the difference is that sampling is done from the *parametric* bivariate distribution

$$N_2\left(\begin{bmatrix} 600.266 \\ 3.095 \end{bmatrix}, \begin{bmatrix} 1746.78 & 7.902 \\ 7.902 & .0593 \end{bmatrix}\right)$$

rather than the *nonparametric* bivariate distribution determined by F_n.

Using 3200 replications, Efron and Tibshirani obtained

$$\widehat{\text{se}(\hat{\rho})}_{\text{parboot}} = .124,$$

which is a little smaller than the nonparametric estimate of .131 that we obtained before. The worse performance of the nonparametric bootstrap can be thought of being the price one pays for having nonparametric assumptions rather than parametric assumptions.

Multiparameter Extensions

In the test score example, we found that the largest relative eigenvalue

was highly significant, but we did not address the significance of the others. In fact, there are 4 parameters in this example: $\theta_1, \theta_2, \theta_3, \theta_4$ where

$$\theta_i = i\text{th largest relative eigenvalue}$$

(We don't need to worry about θ_5 since each of the relative eigenvalues add up to one.)

We can handle such a multiparameter problem through the bootstrap approach with the following generalization of `boot.se()`:

```
########## S function: boot.se ##########

# For obtaining a bootstrap estimate of # the standard error of an
estimator, # defined by the function "estimator" # which takes, as
argument, a matrix "x" # and returns a scalar value.

# Last changed: 13th April, 1998

boot.se <-function(x, num.par, estimator, num.rep=200)
{
   # Convert x to a matrix in case it is an array.
```

```
    x <- as.matrix(x)

    # Set preliminary quantities

    n <- nrow(x)
    estimator.boot <- matrix(0,num.rep,num.par)

    # Obtain bootstrap replications of estimator.

    for(b in 1:num.rep)
    {
       inds.boot <- sample(1:n,n,replace=T)    # sample from {1,..,n} with r'ment.
       x.boot <- x[inds.boot,]                 # obtain bootstrap sample
       estimator.boot[b,] <- estimator(x.boot)# obtain bootstrap estimator
    }

    return(sqrt(diag(var(estimator.boot))))
}

######### End of boot.se ##########
```

We then need the support function:

```
relative.eigenvalues <- function(x)
 {
    eigenvalues <- eigen(var(x))$values

    return(eigenvalues[-ncol(x)]/sum(eigenvalues))
}
```

Standard errors for all 4 parameters can be found using the call:

```
se.ests <- boot.se(testscore.data,4,relative.eigenvalues)
```

We obtain from this the following table:

i	1	2	3	4
$\hat{\theta}_i$	0.62	0.18	0.093	0.076
$\widehat{\text{se}}(\hat{\theta}_i)_{\text{boot}}$	0.049	0.032	0.016	0.01

According to this table, all 4 eigenvalues are, in fact, statistically significant. However, there are two distinctly dominant eigenvalues which suggest that 2 principal components are adequate for the data. The bootstrap can also be used to make inference on the eigenvectors (i.e., the "principal components"). See Efron and Tibshirani for more discussion.

Bootstrap Confidence Intervals

Reading:
A discussion paper by DiCiccio and Efron in the journal *Statistical Science* (1996) contains a lot of interesting contemporary discussion of the current state-of-affairs of bootstrap confidence intervals. We will use this paper as a guideline for the rest of the topic.

So far, we have considered bootstrap confidence intervals of the form

$$\hat{\theta} \pm z_{1-\alpha/2}\widehat{\mathrm{se}(\hat{\theta})}, \tag{1.1}$$

where $\widehat{\mathrm{se}(\hat{\theta})}$ is the bootstrap estimate of the standard error of $\hat{\theta}$.

This interval, of course, comes from the *assumption* that

$$\frac{\hat{\theta} - \theta}{\widehat{\mathrm{se}(\hat{\theta})}} \sim N(0,1).$$

While this assumption is often well-founded, we have seen several examples where bootstrap estimates of the underlying distribution of $\hat{\theta}$

suggest that the normality assumption is far from reasonable.

The bootstrap offers a way of correcting (1.1) by replacing the normality assumption by its own estimate of the sampling distribution of $\hat{\theta}$. As it turns out, there are several (at least 8) ways in which bootstrap replications can be used to construct confidence intervals.

Basic Bootstrap Confidence Intervals

The most obvious approach to getting 90% confidence intervals for θ is to find the upper and lower 5% percentiles of the distribution of the bootstrap replicates $\hat{\theta}_b^*$ and call those a 90% confidence interval for θ. This is known as the *percentile* method and essentially amounts to:

Percentile method for bootstrap confidence intervals

1. Select B independent bootstrap samples

$$\boldsymbol{X}_1^*, \ldots, \boldsymbol{X}_B^*.$$

2. Obtain the estimate of θ for each bootstrap sample:

$$\hat{\theta}_b^* = s(\boldsymbol{X}_b^*), \quad b = 1, \ldots, B.$$

3. Order the $\hat{\theta}_b^*$ values:

$$\hat{\theta}_{(1)}^* \leq \hat{\theta}_{(2)}^* \leq \cdots \leq \hat{\theta}_{(B)}^*.$$

4. The $100(1-\alpha)\%$ confidence interval for θ is

$$(\hat{\theta}_{(\lfloor \alpha B/2 \rfloor)}^*, \hat{\theta}_{(\lfloor (1-\alpha/2)B \rfloor)}^*).$$

Example 1.9

Applying this principle to the problem of getting a 95% confidence interval for the median of the pressure vessel data we get:

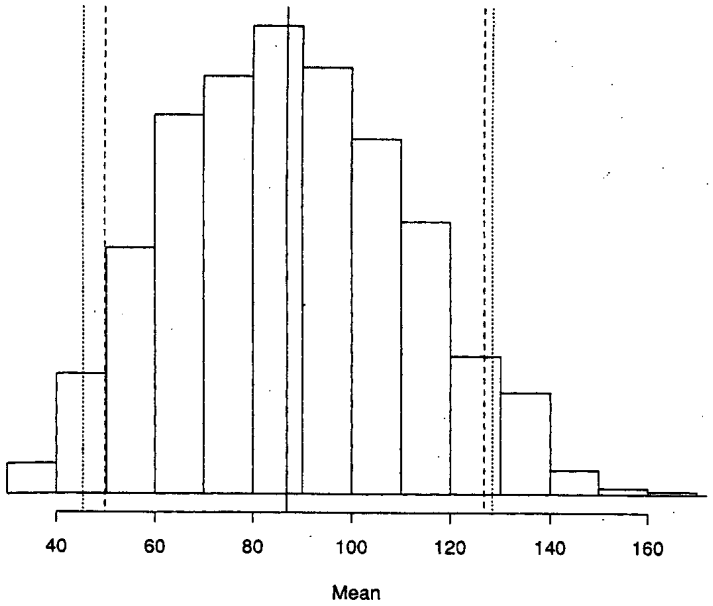
$$(150.5, 849.5).$$

This is somewhat wider than the exact 95% interval based on the exponential model

$$(268.90, 653.10)$$

although it is free of parametric assumptions.

The following figure, taken from Efron and Tibshirani, illustrates that when the bootstrap distribution is approximately normal, the bootstrap confidence interval and the confidence interval based on asymptotic normality give very similar results.



Histogram of 1000 bootstrap replications of $\hat{\theta}$, the mean of the 7 treated mice in Table 2.1. The solid line is drawn at $\hat{\theta}$. The dotted vertical lines show standard normal 90% interval $[86.85 - 1.645 \cdot 25.23, 86.85 + 1.645 \cdot 25.23] = [45.3, 128.4]$. *The dashed vertical lines are drawn at 49.7 and 126.7, the 5% and 95% percentiles of the histogram. Since the histogram is roughly normal-shaped, the broken and dotten lines almost coincide, in accordance with equation (13.2)*

The following `Splus` commands yield percentile confidence limits for the correlation between LSAT and GPA scores in the law school data:

```
temp_bootstrap(x, corr.coeff)
limits,emp(temp)
                2.5%         5%         95%       97.5%
corr.coeff   0.4487901  0.5139906  0.944085  0.9575157
```

This is a definite improvement over the "standard confidence interval" approach, which gave us (.500, 1.05).
Apart from width, we need to worry about *coverage*. It is easy to *claim* that this is a $100(1-\alpha)\%$ confidence interval, but does it really behave the way such an interval should? If $\alpha = 0.05$ then this means that *if the original data are realized many times* then the interval

$$(\hat{\theta}^*_{(\lfloor 0.025B \rfloor)}, \hat{\theta}^*_{(\lfloor 0.975B \rfloor)})$$

should *cover* the true θ about 95% of the time. About the only way to check this is through simulation studies where we know the true value of θ.

Choosing the Number of Bootstrap Replications

For estimation of a standard error or a bias, the general recommendation (e.g. see page 47 of the Efron and Tibshirani book) is that B be in the range

$$25 - 200.$$

Confidence intervals are a much more delicate issue, so there it is recommended that many more replications, about 10 times as much, be performed. The usual range of values for confidence intervals is

$$1000 - 2000.$$

In these notes, we use $B = 2000$ for confidence intervals.
These "rules of thumb" for choices of B will be important in the next section.

Other Simple Bootstrap Confidence Intervals

The percentile method for construction of a confidence interval is the

simplest and "first thing that comes to mind" when one thinks about how to use the bootstrap replications to make a confidence statement. However, it can also be improved. In this section, we will look at some simple ways that try to do this.

The percentile method can be thought of as approximating the distribution of $\hat{\theta}$ by that of $\hat{\theta}^*$, where $\hat{\theta}^*$ represents a typical bootstrap replicate. The rough argument is then:

$$\begin{aligned} 1-\alpha &= P\left(F_{\hat{\theta}}^{-1}\left(\frac{\alpha}{2}\right) \leq \hat{\theta} \leq F_{\hat{\theta}}^{-1}\left(1-\frac{\alpha}{2}\right)\right) \\ &\simeq P\left(F_{\hat{\theta}}^{-1}\left(\frac{\alpha}{2}\right) \leq \theta \leq F_{\hat{\theta}}^{-1}\left(1-\frac{\alpha}{2}\right)\right) \\ &\simeq P\left(F_{\hat{\theta}^*}^{-1}\left(\frac{\alpha}{2}\right) \leq \theta \leq F_{\hat{\theta}^*}^{-1}\left(1-\frac{\alpha}{2}\right)\right). \end{aligned}$$

The first approximation replaces $\hat{\theta}$ by its estimand θ. This seems reasonable since, usually,

$$\hat{\theta} \xrightarrow{P} \theta.$$

The second approximation replaces the distribution of $\hat{\theta}$ by that of $\hat{\theta}^*$. But there are really two approximations going on here, which contribute to coverage error.

A simple way around the first approximation is to work with $\hat{\theta} - \theta$ directly and approximate its distribution with that of $\hat{\theta}^* - \hat{\theta}$. Each of these can be shown to converge to the same thing, namely zero. So this should be more accurate. The confidence interval argument would then be:

$$\begin{aligned} 1-\alpha &= P\left(F^{-1}_{\hat{\theta}-\theta}\left(\frac{\alpha}{2}\right) \leq \hat{\theta}-\theta \leq F^{-1}_{\hat{\theta}-\theta}\left(1-\frac{\alpha}{2}\right)\right) \\ &\simeq P\left(F^{-1}_{\hat{\theta}^*-\hat{\theta}}\left(\frac{\alpha}{2}\right) \leq \hat{\theta}-\theta \leq F^{-1}_{\hat{\theta}^*-\hat{\theta}}\left(1-\frac{\alpha}{2}\right)\right) \\ &= P\left(\hat{\theta}-F^{-1}_{\hat{\theta}^*-\hat{\theta}}\left(1-\frac{\alpha}{2}\right) \leq \theta \leq \hat{\theta}-F^{-1}_{\hat{\theta}^*-\hat{\theta}}\left(\frac{\alpha}{2}\right)\right). \end{aligned}$$

Percentile Method (type-2) for Bootstrap Confidence Intervals

1. Select B independent bootstrap samples

$$\boldsymbol{X}_1^*, \ldots, \boldsymbol{X}_B^*.$$

2. Obtain the estimate of θ for each bootstrap sample:

$$\hat{\theta}_b^* = s(\boldsymbol{X}_b^*), \quad b = 1, \ldots, B$$

and then compute

$$\hat{\phi}_b^* = \hat{\theta}_b^* - \hat{\theta}.$$

3. Order the $\hat{\phi}_b^*$ values:

$$\hat{\phi}_{(1)}^* \leq \hat{\phi}_{(2)}^* \leq \cdots \leq \hat{\phi}_{(B)}^*.$$

4. The $100(1-\alpha)\%$ confidence interval for θ is

$$(\hat{\theta} - \hat{\phi}_{(\lfloor (1-\alpha/2)B \rfloor)}^*, \hat{\theta} - \hat{\phi}_{(\lfloor (\alpha/2)B \rfloor)}^*).$$

A more concrete way to appreciate why the type-2 percentile method might lead to an improvement is to suppose the data are from the $N(\mu, \sigma^2)$ distribution and that

$$\theta = \mu \quad \text{and} \quad \hat{\theta} = \bar{X}.$$

Then it can be shown that

$$\bar{X}^* \overset{approx.}{\sim} N(\bar{X}, S^2/n)$$

while

$$\bar{X} \sim N(\mu, \sigma^2/n).$$

So confidence intervals based on the distribution of $\bar{X}^*$ will suffer because of two possible erroneous approximations: $\bar{X}$ to μ and S^2 to σ^2. On the other hand,

$$\bar{X}^* - \bar{X} \overset{approx.}{\sim} N(0, S^2/n),$$

while

$$\bar{X} - \mu \sim N(0, \sigma^2/n).$$

So the type-2 method only has to live with one approximation: S^2 to σ^2.

How might we get down to *no approximations*? For the normal example, we could work with

$$\frac{\bar{X}^* - \bar{X}}{S^*/\sqrt{n}} \quad \text{as an approximation to} \quad \frac{\bar{X} - \mu}{S/\sqrt{n}}.$$

Then it can be shown that

$$\frac{\bar{X}^* - \bar{X}}{S^*/\sqrt{n}} \overset{approx.}{\sim} t_{n-1},$$

as well as the famous result

$$\frac{\bar{X} - \mu}{S/\sqrt{n}} \sim t_{n-1}.$$

These agree! The bootstrap-t method only has to deal with the bootstrap error, and not any extra estimation error. This can be shown to lead to better asymptotic performance.
For general $\hat{\theta}$, the last idea can be expressed in terms of the distribution

of $\frac{\hat{\theta}^* - \hat{\theta}}{\text{se}(\hat{\theta}^*)}$ approximating the distribution of $\frac{\hat{\theta} - \theta}{\text{se}(\hat{\theta})}$, so that

$$\begin{aligned}
1 - \alpha &= P\left(F^{-1}_{\frac{\hat{\theta} - \theta}{\text{se}(\hat{\theta})}}\left(\frac{\alpha}{2}\right) \leq \frac{\hat{\theta} - \theta}{\text{se}(\hat{\theta})} \leq F^{-1}_{\frac{\hat{\theta} - \theta}{\text{se}(\hat{\theta})}}\left(1 - \frac{\alpha}{2}\right)\right) \\
&\simeq P\left(F^{-1}_{\frac{\hat{\theta}^* - \hat{\theta}}{\text{se}(\hat{\theta}^*)}}\left(\frac{\alpha}{2}\right) \leq \frac{\hat{\theta} - \theta}{\text{se}(\hat{\theta})} \leq F^{-1}_{\frac{\hat{\theta}^* - \hat{\theta}}{\text{se}(\hat{\theta}^*)}}\left(1 - \frac{\alpha}{2}\right)\right) \\
&= P\left(\hat{\theta} - \text{se}(\hat{\theta}) F^{-1}_{\frac{\hat{\theta}^* - \hat{\theta}}{\text{se}(\hat{\theta}^*)}}\left(1 - \frac{\alpha}{2}\right) \leq \theta \leq \hat{\theta} - \text{se}(\hat{\theta}) F^{-1}_{\frac{\hat{\theta}^* - \hat{\theta}}{\text{se}(\hat{\theta}^*)}}\left(\frac{\alpha}{2}\right)\right).
\end{aligned}$$

This results in the *bootstrap-t* method (also known as the *percentile-t* method):

Bootstrap-t method for confidence intervals

1. Select B independent bootstrap samples

$$\boldsymbol{X}_1^*, \ldots, \boldsymbol{X}_B^*.$$

2. Obtain the estimate of θ for each bootstrap sample:

$$\hat{\theta}_b^* = s(\boldsymbol{X}_b^*), \quad b = 1, \ldots, B$$

and $\text{se}(\hat{\theta}_b^*) = \text{se}(s(\boldsymbol{X}_b^*))$. Then compute

$$\hat{T}_b^* = \frac{\hat{\theta}_b^* - \hat{\theta}}{\text{se}(\hat{\theta}_b^*)}.$$

3. Order the $\hat{T}_b^*$ values:

$$\hat{T}_{(1)}^* \leq \hat{T}_{(2)}^* \leq \cdots \leq \hat{T}_{(B)}^*.$$

4. The $100(1-\alpha)\%$ confidence interval for θ is

$$(\hat{\theta} - \text{se}(\hat{\theta})\hat{T}_{(\lfloor(1-\alpha/2)B\rfloor)}^*, \hat{\theta} - \text{se}(\hat{\theta})\hat{T}_{(\lfloor(\alpha/2)B\rfloor)}^*)$$

.

There is a subtle issue here that is, perhaps, not apparent on first reading: the requirement of a formula for

$$\text{se}(\hat{\theta}).$$

For the $\bar{X}$ problem this is easy:

$$\text{se}(\bar{X}) = \frac{S}{\sqrt{n}}.$$

But for a general statistic $\hat{\theta}$ we don't always have this. Of course, the previous section was devoted to estimation of

$$\text{se}(\hat{\theta})$$

through the use of the bootstrap, but the problem with the bootstrap-t interval is that we need to find

$$\text{se}(\hat{\theta}_b^*)$$

for each $b = 1, \ldots, B$. In other words, we need to do two *nested* bootstraps! So given that a bootstrap standard error requires at least 25 replications and that confidence intervals require at least 1000 replications, this means that we need at least

$$25,000 \text{ replications}$$

to use the bootstrap-t method in those cases where we don't have a

standard error formula! For those problems that require 200 standard error replications and 2000 confidence interval replications we end up with

$$400,000 \text{ replications!}$$

So the computational burden of the "fully bootstrapped" t-statistic approach to confidence interval construction can be prohibitive.

Nevertheless, there are many problems where standard error rules exist. For the correlation coefficient, it can be argued that the standard error has the reasonable method of moments estimate:

$$\text{se}(\hat{\rho}) = \sqrt{\frac{\hat{\rho}^2}{n}\left\{\frac{\hat{\mu}_{22}}{\hat{\mu}_{11}^2} + \frac{1}{4}\left(\frac{\hat{\mu}_{40}}{\hat{\mu}_{20}^2} + \frac{\hat{\mu}_{04}}{\hat{\mu}_{02}^2} + \frac{2\hat{\mu}_{22}}{\hat{\mu}_{20}\hat{\mu}_{02}}\right) - \left(\frac{\hat{\mu}_{31}}{\hat{\mu}_{20}\hat{\mu}_{11}} + \frac{\hat{\mu}_{13}}{\hat{\mu}_{02}\hat{\mu}_{11}}\right)\right\}},$$

where

$$\hat{\mu}_{ij} = \frac{1}{n}\sum_{k=1}^{n}(X_k - \bar{X})^i(X_k - \bar{X})^j.$$

This is messy, but it sure beats having to do nested bootstrap replications.

How do these various approaches compare? In the 1980's, a lot of research was devoted to the asymptotic comparison of these approaches – led mainly by Peter Hall of the Australian National University. It shows that, for example, the bootstrap-t method is asymptotically superior than the percentile method in that its error is

$$O_p(n^{-1})$$

while the percentile method has an error of

$$O_p(n^{-1/2}).$$

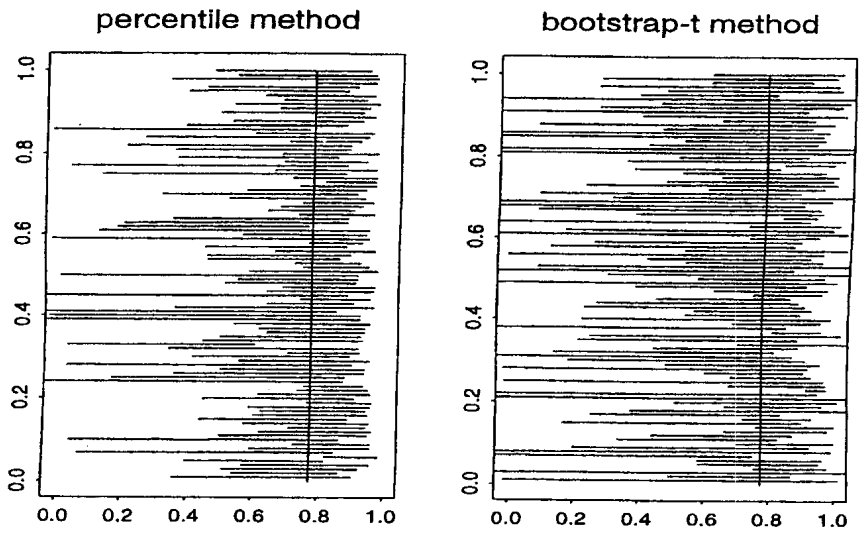
Despite these asymptotic results, it is small sample performance that is most relevant to practice and about the only way to understand this is through simulation. A simulation study for inference about the correlation coefficient ρ was conducted in which the data were assumed to follow a

$$N_2\left(\begin{bmatrix} 600.266 \\ 3.095 \end{bmatrix}, \begin{bmatrix} 1746.78 & 7.902 \\ 7.902 & .0593 \end{bmatrix}\right)$$

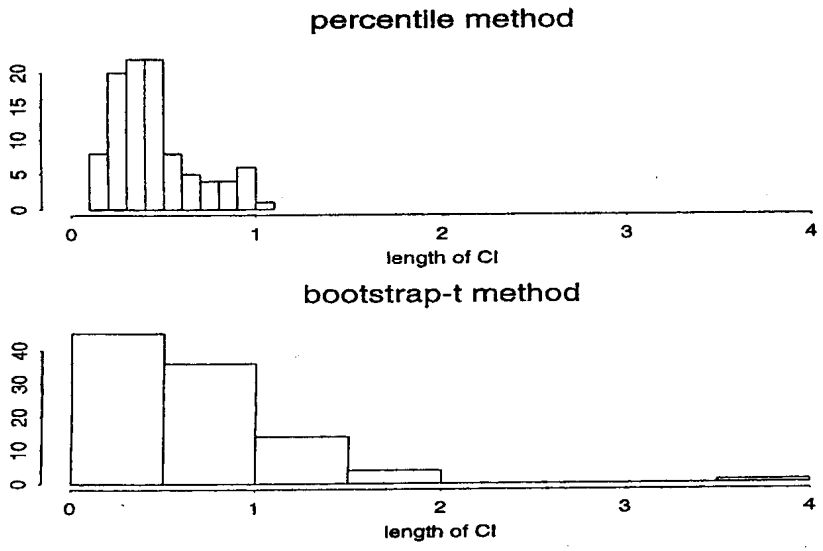
distribution. The parametric fit for the law school data was run. One hundred bivariate samples were simulated from this distribution, where it is known that the true correlation is

$$\rho = 0.776.$$

For each sample, 95% confidence intervals were computed using the percentile method and the bootstrap-t method. The following Figure shows the intervals for the two methods.


percentile method
bootstrap-t method
0.0
0.2
0.4
0.6
0.8
1.0

In terms of coverage both perform about the same. The percentile method covers the true value in 94 out of 100 cases, while the bootstrap-t method covers ρ in 95 out of 100 cases. How about length? The following Figure shows histograms of the lengths of the intervals.


percentile method
0 5 10 15 20
0 1 2 3 4
length of CI
bootstrap-t method
0 10 20 30 40
0 1 2 3 4
length of CI

These show that the percentile method tends to outperform the bootstrap-t method in terms of having a tighter interval more often. Nevertheless, both methods tend to produce very wide intervals, often larger than 1. One of the bootstrap-t intervals had length equal to 3.74! This example shows that, while the bootstrap-t method has theoretical advantages, it does not necessarily perform as well in practice. This is due to the instability of the estimate of $\text{se}(\hat{\rho})$.

It would take many more simulation studies to fully appreciate the practical pros and cons of these bootstrap confidence interval methods. Michael Martin of the Australian National University has performed a lot of such studies, which are summarized in the following Table.

Table 6: Practical performance of simple bootstrap confidence interval methods

	percentile	bootstrap-t
pros	easy to calculate; stable easy to understand	good coverage; works well if good standard error estimate available
cons	poor coverage	unstable when standard error estimate is unstable, not range-preserving, harder to calculate and understand

The methods described in this section, while simple, do have their drawbacks in terms of practical performance. To get more consistently good performance, more clever and subtle methods for confidence interval construction are required.

The next few topics:

1. Cross Validation.
2. Bootstrap Hypothesis tests.

Cross Validation

Reading:

Gong, G. (1986). Cross-Validation, the Jackknife, and the Bootstrap: Express Error Estimation in Forward Logistic Regression. *Journal of the American Statistical Association,* Vol. 81, pp. 108-113.
Efron and Tibshirani, Chapters 17, 16 and 24.

Statisticians are often interested in *Prediction Error*, which measures how well a model predicts the response value of a future observation. Cross-validation is a standard tool for calculating prediction error. The idea actually predates the bootstrap, but has enjoyed a resurgence in recent years due to the availability of sufficient computing power to apply the method.

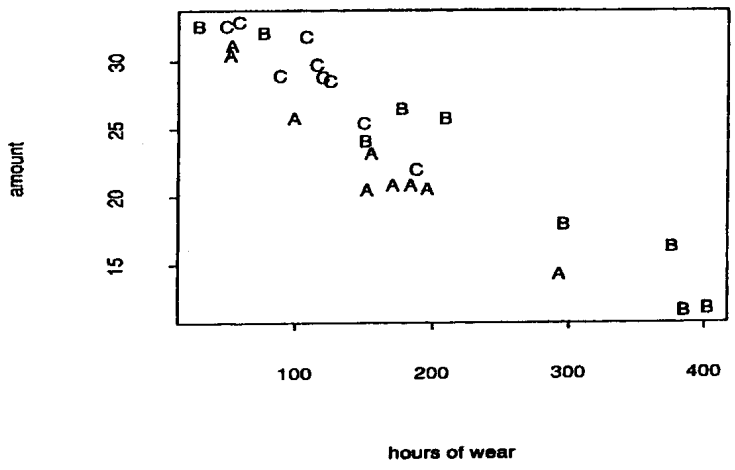
In regression models, prediction error refers to the squared difference

between a future observation and its prediction based on the model:

$$\text{PE} = E(\boldsymbol{y} - \hat{\boldsymbol{y}})^2.$$

Example 1.10

Efron and Tibshirani present a dataset involving the amount of hormone remaining in a device after various hours of wear. The devices come from three different lots (A, B and C). The question of interest is how well the model will predict the amount of hormone remaining for a new device.


amount
15
20
25
30
100
200
300
400
hours of wear

Note that the definition can be applied for any kind of regression model, linear or non-linear, normal, logistic, etc...

One of the most common measures for the "goodness of fit" of a regression curve to a scatterplot is the *residual squared error*, given by

$$\text{RSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \text{RSS}/n.$$

However, this will tend to be too optimistic, since we are using the same data to both fit and assess the model.

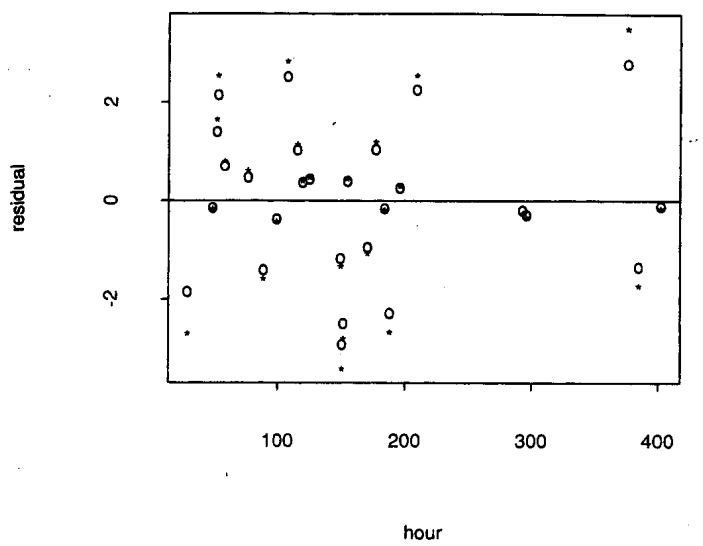
While one approach is to replace n with $n - p$, it turns out that a bigger correction is needed.

A naive approach sometimes used in practice is to divide the sample in half, use the first half to fit the model and the second half to validate. However, this approach is very inefficient. *Cross-validation* is a classical "resampling-type" version of this same idea, but it uses the data more more

efficiently. The *Cross-validation* criterion is

$$\text{CV} = \frac{1}{n}\sum_{i=1}^{n}\{y_i - \hat{y}_i^{(-i)}\}^2,$$

where $\hat{y}_i^{(-i)}$ denotes the regression estimator applied to the data, but with $(\boldsymbol{x}_i, y_i)$ deleted. This "leaving-one-out" strategy is a way of guarding against the "interpolation" answer that RSE gives.
When applied to the hormone data, the value of CV turns out to be 3.09 (compared to 2.20 for RSE). The following figure shows the usual residuals (circles) and cross validated residuals (stars).



Plot of residuals (circles) and cross-validated residuals (stars) for hormone data.

Shao and Tu discuss some advantages of a more general version of cross validation, breaking the data into K parts:

K-fold cross-validation

1. Split the data into K roughly equal parts.
2. For the kth part, fit the model to the other $K-1$ parts of the data and calculate the prediction error of the fitted model when predicting the kth part of the data.
3. Repeat for $k = 1, 2, \cdots, K$ and combine the K estimates of the prediction error.

On the surface, it appears that Cross-validation requires refitting the model K times. In practice, however, there are some useful analytic approximations which are exact in the case of linear regression.

Example 1.11 (Linear Regression)

Suppose we observe $(\boldsymbol{x}_1, y_1), \cdots, (\boldsymbol{x}_n, y_n)$ and want to fit

$$y_i = \boldsymbol{x}_i' \boldsymbol{\beta} + \epsilon_i.$$

The ordinary least squares regression fit to these data is

$$\hat{\boldsymbol{y}} = \boldsymbol{X}\hat{\boldsymbol{\beta}} \quad \text{where} \quad \hat{\boldsymbol{\beta}} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y} \tag{1.2}$$

and

$$\hat{\boldsymbol{y}} = \begin{bmatrix} \hat{y_1} \\ \vdots \\ \hat{y_n} \end{bmatrix}$$

is the vector of *fitted values*. We can rewrite (1.2) as

$$\hat{\boldsymbol{y}} = \boldsymbol{H}\boldsymbol{y},$$

where

$$\boldsymbol{H} = \boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'$$

is an $n \times n$ matrix commonly known as the *"hat" matrix*.

It follows that

$$y_i - \hat{y}_i^{(-i)} = \frac{y_i - \hat{y}_i}{1 - h_{ii}},$$

where h_{ii} is the ith diagonal element of $\boldsymbol{H}$, leading to

$$\text{CV} = \sum_{i=1}^{n} \left\{ \frac{y_i - \hat{y}_i}{1 - h_{ii}} \right\}^2.$$

Even this last expression can be tedious to calculate. A popular simplified version replaces h_{ii} by $\bar{h} = \frac{1}{n} \sum_{i=1}^{n} h_{ii}$, resulting in the so-called *Generalized Cross Validation* estimator of the prediction error:

$$\text{CV} = \sum_{i=1}^{n} \left\{ \frac{y_i - \hat{y}_i}{1 - \bar{h}} \right\}^2 = \text{RSS}/(1 - \bar{h})^2.$$

If we note that $\sum_{i=1}^{n} h_{ii} = \text{tr}(\boldsymbol{H})$, then we get a nice simplification in the

case of ordinary least squares,

$$\begin{aligned} \text{tr}(\boldsymbol{H}) &= \text{tr}\left\{\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\right\} \\ &= \text{tr}\left\{(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{X}\right\} \\ &= \text{tr}(\boldsymbol{I}_p) \\ &= p, \end{aligned}$$

which is the number of parameters fitted in the regression model, often called the *degrees of freedom* of the fit.

Application to Model Selection

Estimators of model prediction error can be used as the basis for selecting from among a set of models.

One can think of GCV as being an augmentation of RSS, where the presence of $\text{tr}(\boldsymbol{H})$ on the denominator acts as a type of penalty for having too many degrees of freedom. It turns out that there are several other such model selection criteria that combine RSS and $\text{tr}(\boldsymbol{H})$ in various analogous ways.

Mallows' C_p and other Criteria

Mallows developed the following measure in the early 1970's:

$$C_p = \text{RSS}/n + 2p\hat{\sigma}^2/n,$$

where p is the number of parameters in the candidate model, RSS is the residual sum of squares, and

$$\hat{\sigma}^2 = \frac{\text{RSS}}{n-p}$$

is the estimate of residual variance based on the largest possible model being considered. The idea is then to choose the model with the smallest C_p value. When C_p is being used to choose between a set of potential regression models, $\hat{\sigma}^2$ is usually calculated from the largest (richest) model.

It can be shown that $C_p \approx \text{E(PE)}$.

Mallows' C_p is a special case of Akiake's Information Criterion (AIC)

which applies for more general models:

$$\text{AIC} = -2\log L(\hat{\boldsymbol{\theta}}) + 2p,$$

where $L(\boldsymbol{\theta})$ is the likelihood function of $\boldsymbol{\theta}$.
A related quantity is the *Bayesian Information Criterion* (BIC):

$$\text{BIC} = -2\log L(\hat{\boldsymbol{\theta}}) + \log(n)p.$$

Because BIC replaces the 2 from C_p with $\log(n)$, it provides a more severe penalty and tends to favor more parsimonious models.

In contrast to both adjusted RSS and C_p, BIC is a *consistent criterion* in that it chooses the correct model as $n \to \infty$.

Shao and Tu provide some interesting discussion on this topic and show the results of a simulation based on a linear regression model with 5 predictors, a sample size of 40, and iid standard normal errors.

Table 9: Empirical model selection probabilities

True $\boldsymbol{\theta}$	α	AIC	C_p	BIC	$\text{CV}_1^\dagger$	$\text{CV}_{25}^\dagger$
(2,0,0,4,0)	1, 4‡	.567	.594	.804	.484	.934
	1,2,4	.114	.110	.049	.133	.025
	1,3,4	.126	.113	.065	.127	.026
	1,4,5	.101	.095	.057	.138	.012
	1,2,3,4	.030	.029	.009	.049	.000
	1,2,4,5	.030	.027	.007	.029	.001
	1,3,4,5	.022	.026	.008	.030	.002
	1,2,3,4,5	.010	.007	.001	.009	.000

Table 9 (continued): Empirical model selection probabilities

True $\boldsymbol{\theta}$	α	AIC	C_p	BIC	$\text{CV}_1^\dagger$	$\text{CV}_{25}^\dagger$
(2,0,0,4,8)	1, 4, 5‡	.683	.690	.881	.641	.947
	1,2,4,5	.143	.129	.045	.158	.032
	1,3,4,5	.116	.142	.067	.138	.020
	1,2,3,4,5	.058	.039	.007	.063	.001
(2,9,0,4,8)	1,4,5	.000	.000	.000	.005	.016
	1, 2, 4, 5‡	.794	.817	.939	.801	.965
	1,3,4,5	.000	.000	.000	.005	.002
	1,2,3,4,5	.206	.183	.061	.189	.017
(2,9,6,4,8)	1,4,5	.000	.000	.000	.000	.002
	1,2,4,5	.000	.000	.000	.000	.005
	1,3,4,5	.000	.000	.000	.015	.045
	1, 2, 3, 4, 5‡	1.00	1.00	1.00	.985	.948

† CV_1 stands for the delete-d CV.
‡ The optimal model.

Why bother with real cross validation when these approximations are available? Several reasons:

- Approximations are only that.
- Degrees of freedom are not always clear (e.g., CART example in Efron and Tibshirani; nonparametric smoothing settings).
- Robustness.

The bootstrap can provide a more computationally appealing approach than cross validation.

A simplistic bootstrap estimate of prediction error

1. Select B independent bootstrap samples: $(\boldsymbol{y}_b^*, \boldsymbol{X}_b^*), \quad b = 1, \cdots, B.$
2. Let $\hat{\boldsymbol{\beta}}_b^*$ be the regression coefficients obtained from fitting the model to the bth bootstrapped sample.

3. Compute
$$\frac{1}{n}\sum_{i=1}^{n}(y_i - \boldsymbol{x}_i'\hat{\boldsymbol{\beta}}_b^*)^2.$$
4. Repeat steps 2 and 3 for $b = 1, 2, \cdots, B$, then average the results.

This simplistic approach turns out not to work well. Why? Basically, the estimate of the RSE is not consistent when based on an incorrect model (which will often be the case when we are considering model selection). The following table (from Efron and Tibshirani) shows the estimated prediction errors based on 10 bootstrapped samples from the hormone data. The first column displays the estimated prediction error obtained for each sample using the method outlined above.

The second column shows the (low) estimate that results when the procedure is applied to the bootstrapped sample itself. The difference between the two columns (called *optimism*) shows the difference and is the amount by which the average residual squared error underestimates the true prediction error.

	$\text{err}(\boldsymbol{x}^*, \hat{F})$	$\text{err}(\boldsymbol{x}^*, \hat{F}^*)$	$\text{err}(\boldsymbol{x}^*, \hat{F})$ - $\text{err}(\boldsymbol{x}^*, \hat{F}^*)$
sample 1:	2.30	1.47	.83
sample 2:	2.56	3.03	-0.47
sample 3:	2.30	1.65	.65
sample 4:	2.43	1.78	.67
sample 5:	2.44	2.00	.44
sample 6:	2.67	1.17	1.50
sample 7:	2.68	1.23	1.45
sample 8:	2.39	1.55	.84
sample 9:	2.86	1.76	1.10
sample 10:	2.54	1.37	1.17
AVERAGE	2.52	1.70	.82

Take bootstrap sample 9, for example. The value 2.86 is the result of applying Step 3 from the "simplistic bootstrap estimate" to this example. The value 1.76 is a "naive" estimate of the prediction error obtained by

$$\frac{1}{n}\sum_{i=1}^{n}(y_{ib}^{*} - \boldsymbol{x}_{ib}^{*'}\hat{\boldsymbol{\beta}}_{b}^{*})^{2},$$

where y_{ib}^{*} and $\boldsymbol{x}_{ib}^{*}$ are the ith observation and its corresponding covariate in the bth bootstrap sample, and $\hat{\boldsymbol{\beta}}_{b}^{*}$ is the estimated regression coefficient based on the bth sample. The difference, 2.86-1.76=1.10, is an estimate of how much the naive approach underestimates the true PE. We can exploit this to come up with a better estimate.

A refined bootstrap estimate of prediction error

1. Select B independent bootstrap samples: $(\boldsymbol{y}_b^*, \boldsymbol{X}_b^*), \quad b = 1, \cdots, B.$
2. Let $\hat{\boldsymbol{\beta}}_b^*$ be the regression coefficients obtained from fitting the model to the bth bootstrapped sample.
3. Compute

$$\omega_b = \frac{1}{n}\sum_{i=1}^{n}(y_i - \boldsymbol{x}_i'\hat{\boldsymbol{\beta}}_b^*)^2 - \frac{1}{n}\sum_{i=1}^{n}(y_i^* - \boldsymbol{x}_i^{*\prime}\hat{\boldsymbol{\beta}}_b^*)^2.$$

4. Repeat steps 2 and 3 for $b = 1, 2, \cdots, B$, then average the results to obtain $\hat{\omega} = (1/B)\sum \omega_b$.
5. Estimate the model prediction error by:

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - \boldsymbol{x}_i'\hat{\boldsymbol{\beta}}^*)^2 + \hat{\omega},$$

where $\hat{\boldsymbol{\beta}}^* = (1/B)\sum \hat{\boldsymbol{\beta}}_b^*$.

Bootstrapped Hypothesis Testing

Suppose we want to test the null hypothesis $H_0 : F = G$ and suppose that we have identified a suitable test statistic, $T(X)$, which is a function of the observed data X which comprises n observations from population F and m observations from population G. The test statistic could correspond to an estimate of some parameter that characterizes the difference between the two populations or it could be a score test, a rank test, or anything.

To perform inference, we need to find an *Achieved Significance Level:*

$$\text{ASL} = P_{H_0}\{T(X^*) \geq T(X)\},$$

where $T(X)$ is the observed test statistic and $T(X^*)$ is the test statistic applied to the random variable X^* which has a distribution specified by the null hypothesis. Our usual bootstrap logic applies and suggests the following algorithm:

Bootstrap test for $H_0 : F = G$

1. Select B independent bootstrap samples of size $n + m$ with replacement from X. Arbitrarily call the first n observations $\boldsymbol{z}$ and the second m $\boldsymbol{y}$.
2. Evaluate $T(\boldsymbol{x}_b^*)$, for example,

$$T(\boldsymbol{x}_b^*) = \bar{\boldsymbol{z}}^* - \bar{\boldsymbol{y}}^*, \quad b = 1, \cdots, B.$$

3. Approximate ASL_{BOOT} by

$$\widehat{\text{ASL}}_{\text{BOOT}} = \#\{T(\boldsymbol{x}_b^*) \geq T(\boldsymbol{x})\}/B,$$

where $T(\boldsymbol{x})$ is the observed test statistic.

This procedure is very similar to a permutation test, except that sampling is with replacement.