Bootstrap Estimation

Suppose a simple random sample (sampling with replacement)

$$\mathrm{X}_1, \mathrm{X}_2, \ldots, \mathrm{X}_n$$

is available from some population with distribution function (cdf)

$$F(\mathrm{x}).$$

Objective: Make inferences about some feature of the population

- median
- variance
- correlation

A statistic t_n is computed from the observed data:

Sample mean: $t_n = \frac{1}{n} \sum_{j=1}^{n} X_j$

Standard deviation:

$$t_n = \sqrt{\frac{1}{n-1} \sum_{j=1}^{n} (X_j - \overline{X})^2}$$

Correlation:

$$\mathrm{X}_j = \begin{bmatrix} X_{1j} \\ X_{2j} \end{bmatrix} \qquad j = 1, 2, \ldots, n$$

$$t_n = \frac{\sum_{j=1}^{n} (X_{1j} - \overline{X}_{1.})(X_{2j} - \overline{X}_{2.})}{\sqrt{\sum_{j=1}^{n} (X_{1j} - \overline{X}_{1.})^2} \sqrt{\sum_{j=1}^{n} (X_{2j} - \overline{X}_{2.})^2}}$$

t_n estimates some feature of the population.

What can you say about the distribution of t_n, with respect to all possible samples of size n from the population?

- Expectation of t_n
- Standard deviation
- Distribution function

How can a confidence interval be constructed?

Simulation: (The population c.d.f. is known)

- For a univariate normal distribution with mean μ and variance σ^2, the *cdf* is

$$\begin{aligned} F(x) &= Pr\{X \leq x\} \\ &= \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}(\frac{w-\mu}{\sigma})^2} dw \end{aligned}$$

- Simulate B samples of size n and compute the value of t_n for each sample:

$$t_{n,1}, t_{n,2}, \ldots, t_{n,B}$$

- Approximate $E_F(t_n)$ with the simulated mean

$$\bar{t} = \frac{1}{B} \sum_{k=1}^{B} t_{n,k}$$

- Approximate $Var_F(t_n)$ with

$$\frac{1}{B-1} \sum_{k=1}^{B} (t_{n,k} - \bar{t})^2$$

- Approximate the standard deviation of t_n with

$$\sqrt{\frac{1}{B-1} \sum_{k=1}^{B} (t_{n,k} - \bar{t})^2}$$

- Approximate the c.d.f. for t_n with

$$\bar{F}_n(t) = \frac{\text{number of samples with } t_{n,k} < t}{B}$$

- Order the B values of t_n from smallest to largest

$$t_{n(1)} \leq t_{n(2)} \leq \cdots \leq t_{n(B)}$$

and approximate percentiles of the distribution of t_n.

What if $F(\mathrm{X})$, the population c.d.f. is unknown?

- You cannot use a random number generator to simulate samples of size n and values of t_n, from the actual population.
- Use a bootstrap (or resampling) method?

Basic idea:

(1) Approximate the population cdf

$$F(x)$$

with the empirical cdf

$$\hat{F}_n(x)$$

obtained from the observed sample

$$X_1, X_2, \ldots, X_n$$

Assign probability $\frac{1}{n}$ to each observation in the sample. Then

$$\bar{F}_n(x) = \frac{b}{n}$$

where b is the number of observations in the sample with

$$X_i < x \quad \text{for} \quad i = 1, 2, \ldots, n$$

- Approximate the act of simulating B samples of size n from a population with c.d.f. $F(x)$ by simulating B samples of size n from a population with c.d.f. $\hat{F}_n(x)$
  - The "approximating" population is the original sample.
  - Sample n observations from the original sample using simple random sampling with replacement.

    This will be called a boostrap sample.

  - repeat this B times to obtain B bootstrap samples of size n.
  - Evaluate the summary statistic for each bootstrap sample

    Sample 1: $t^*_{n,1}$

    Sample 2: $t^*_{n,2}$

    ⋮

    Sample B: $t^*_{n,B}$

- Evaluate bootstrap estimates of features of the sampling distribution for t_n, when the c.d.f. is $\hat{F}_n(\mathrm{x})$.

$$E_{\hat{F}_n}(t_n) = \frac{1}{B}\sum_{b=1}^{B} t^*_{n,b}$$

$$Var_{\hat{F}_n}(t_n) = \frac{1}{B-1}\Sigma_{b=1}^{B}\left[t^*_{n,b} - E_{\hat{F}_n}(t_n)\right]^2$$

- This resampling procedure is called a *nonparametric* bootstrap
- If used properly it provides consistent large sample results: As $n \to \infty$ and $B \to \infty$

$$\begin{aligned} E_{\hat{F}_n}(t_n) &= \frac{1}{B}\sum_{b=1}^{B} t^*_{n,b} \\ &\to E_F(t_n) \end{aligned}$$

$$\begin{aligned} \widehat{Var}_{\hat{F}_n}(t_n) &= \frac{1}{B-1}\sum_{b=1}^{B}\left[t^*_{n,b} - \frac{1}{B}\sum_{b=1}^{B} t^*_{n,b}\right]^2 \\ &\to Var_F(t_n) \end{aligned}$$

The bootstrap is a *large sample* method

- Large number of bootstrap samples. As $B \to \infty$

$$\widehat{E_{\hat{F}_n}}(t_n) = \frac{1}{B}\sum_{b=1}^{B} t^*_{n,b} \to E_{\hat{F}_n}(t_n)$$

$$\widehat{Var}_{\hat{F}_n}(t_n) = \frac{1}{B-1}\sum_{b=1}^{B}(t^*_{n,b} - \widehat{E_{\hat{F}_n}}(t_n))^2 \to Var_{\hat{F}_n}(t_n)$$

What is a good value for B?

standard deviation: $B \approx 200$
confidence interval: $B \approx 1000$
more demanding applications: $B \approx 5000$

- Consistency: Original sample size must become large

As $n \to \infty$,

$$\hat{F}_n(\mathrm{x}) \to F(\mathrm{x}) \text{ for any } \mathrm{x}$$

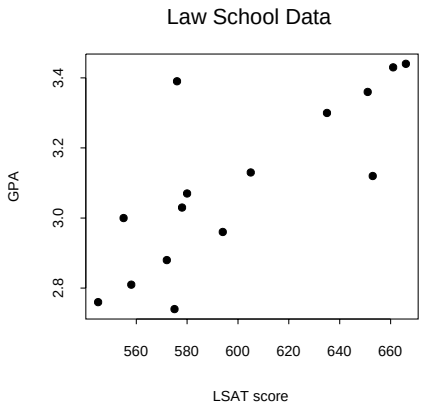
Then,

$$\begin{aligned} E_{\hat{F}_n}(t_n) &\to E_F(t_n) \\ Var_{\hat{F}_n}(t_n) &\to Var_F(t_n) \\ \vdots & \quad \vdots \end{aligned}$$

- For small values of n, $\hat{F}_n(\mathrm{x})$ could deviate substantially from $F(\mathrm{x})$.

Example 12.1: Average values for GPA and LSAT scores for students admitted to n=15 Law Schools in 1973.

School	LSAT	GPA
1	576	3.39
2	635	3.30
3	558	2.81
4	579	3.03
5	666	3.44
6	580	3.07
7	555	3.00
8	661	3.43
9	651	3.36
10	605	3.13
11	653	3.12
12	575	2.74
13	545	2.76
14	572	2.88
15	594	2.96



- These schools were randomly selected from a larger population of law schools.
- We want to make inferences about the correlation between GPA and LSAT scores for the population of law schools
- The sample correlation (n=15) is

$$r = 0.7764 \equiv t_n$$

Bootstrap samples

Take samples of n=15 schools, using simple random sampling with replacement

Sample 1:

School	LSAT	GPA
7	555	3.00
10	605	3.13
14	572	2.88
8	661	3.43
5	666	3.44
4	578	3.03
1	576	3.39
2	635	3.30
13	545	2.76
3	558	2.81
15	594	2.96
4	573	3.03
9	651	3.36
3	558	2.81
3	558	2.81

$$t^*_{n,1} = 0.8586 = r_{15,1}$$

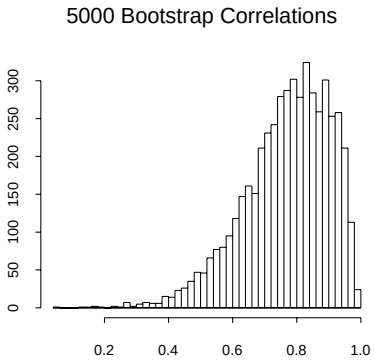
Sample 2:

School	LSAT	GPA
12	575	2.74
11	653	3.12
12	575	2.74
8	661	3.43
9	651	3.36
5	666	3.44
9	651	3.36
1	576	3.39
5	666	3.44
5	666	3.44
1	576	3.39
2	635	3.30
10	605	3.13
14	572	2.88

$$t^*_{n,2} = 0.6673 = r_{15,2}$$

Repeat this $B = 5000$ times to obtain

$$t^*_{n,1},\ t^*_{n,2},\ \ldots,\ t^*_{n,5000}$$



Estimated correlation from the original sample of $n = 15$ law schools

$$r = 0.7764$$

Bootstrap standard error (from $B = 5000$ bootstrap samples) is

$$S_r = \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}\left[t^*_{n,b} - \frac{1}{B}\sum_{j=1}^{B} t^*_{n,j}\right]^2}$$

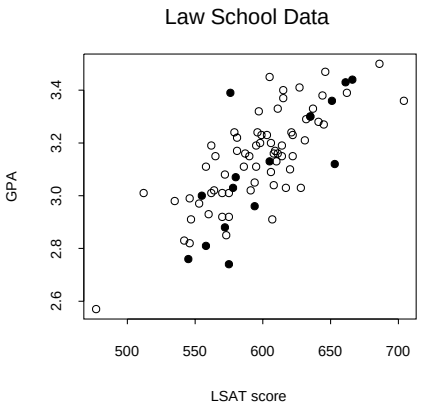
$$= \sqrt{\frac{1}{4999}\sum_{b=1}^{5000}\left[r_{15,b} - \frac{1}{5000}\sum_{j=1}^{5000} r_{15,j}\right]^2}$$

$$= 0.1341$$

Number of Bootstrap samples	Bootstrap estimate of Standard error
$B = 25$	0.1108
$B = 50$	0.0985
$B = 100$	0.1334
$B = 250$	0.1306
$B = 500$	0.1328
$B = 1000$	0.1299
$B = 2500$	0.1366
$B = 5000$	0.1341

"Very seldom are more than $B = 200$ replications needed for estimating a standard error."

– **Efron & Tibshirani (1993) (page 52)**



	School	LSAT	GPA	Type
1	1	622	3.23	2
2	2	542	2.83	2
3	3	579	3.24	2
4	4	653	3.12	1
5	5	606	3.09	2
6	6	576	3.39	1
7	7	620	3.10	2
8	8	615	3.40	2
9	9	553	2.97	2
10	10	607	2.91	2
11	11	558	3.11	2
12	12	596	3.24	2
13	13	635	3.30	1
14	14	581	3.22	2
15	15	661	3.43	1
16	16	547	2.91	2
17	17	599	3.23	2
18	18	646	3.47	2
19	19	622	3.15	2
20	20	611	3.33	2
21	21	546	2.99	2

	School	LSAT	GPA	Type
22	22	614	3.19	2
23	23	628	3.03	2
24	24	575	3.01	2
25	25	662	3.39	2
26	26	627	3.41	2
27	27	608	3.04	2
28	28	632	3.29	2
29	29	587	3.16	2
30	30	581	3.17	2
31	31	605	3.13	1
32	32	704	3.36	2
33	33	477	2.57	2
34	34	591	3.02	2
35	35	578	3.03	1
36	36	572	2.88	1
37	37	615	3.37	2
38	38	606	3.20	2
39	39	603	3.23	2
40	40	535	2.98	2
41	41	595	3.11	2
42	42	575	2.92	2

	School	LSAT	GPA	Type
43	43	573	2.85	2
44	44	644	3.38	2
45	45	545	2.76	1
46	46	645	3.27	2
47	47	651	3.36	1
48	48	562	3.19	2
49	49	609	3.17	2
50	50	555	3.00	1
51	51	586	3.11	2
52	52	580	3.07	1
53	53	594	2.96	1
54	54	594	3.05	2
55	55	560	2.93	2
56	56	641	3.28	2
57	57	512	3.01	2
58	58	631	3.21	2
59	59	597	3.32	2
60	60	621	3.24	2
61	61	617	3.03	2
62	62	637	3.33	2
63	63	572	3.08	2

	School	LSAT	GPA	Type
64	64	610	3.13	2
65	65	562	3.01	2
66	66	635	3.30	2
67	67	614	3.15	2
68	68	546	2.82	2
69	69	598	3.20	2
70	70	666	3.44	1
71	71	570	3.01	2
72	72	570	2.92	2
73	73	605	3.45	2
74	74	565	3.15	2
75	75	686	3.50	2
76	76	608	3.16	2
77	77	595	3.19	2
78	78	590	3.15	2
79	79	558	2.81	1
80	80	611	3.16	2
81	81	564	3.02	2
82	82	575	2.74	1

The correlation coefficient for the population of 82 Law Schools in 1973 is

$$\rho = 0.7600$$

The exact distribution of estimated correlation coefficients for random samples of size n=15 from this population involves

More than 3.6×10^{110} possible samples of size n=15

Results from 100,000 samples of size n=15.

Percentiles	
.95	0.9128
.75	0.8404
.50	0.7699
.25	0.6777
.05	0.5010

min = -0.3647

max = 0.9856

mean = 0.7464

std. error = 0.1302

Bias: From a sample of size n, t_n is used to estimate a population parameter θ.

$$Bias_F(t_n) = E_F(t_n) - \theta$$

$E_F(t_n)$: average across all possible samples of size n from the population

θ: true parameter value

Law School example:

$$\begin{aligned} Bias_F(r_{15}) &= E_F(r_{15}) - 0.7600 \\ &= .7464 - .7600 \\ &= -0.0136 \end{aligned}$$

Bootstrap estimate of bias:

"true value" if you take the original sample as the population

$$Bias_{\hat{F}}(t_n) = E_{\hat{F}}(t_n) - t_n$$

t_n: approximate this with the average of results from B bootstrap samples $\frac{1}{B}\sum_{b=1}^{B} t^*_{n,b}$

Law School example:

($B = 5000$ bootstrap samples)

$$\begin{aligned} \widehat{Bias}_{\bar{F}}(r_{15}) &= \frac{1}{5000}\Sigma_{b=1}^{5000} r_{15,b} - r_{15} \\ &= .7692 - .7764 \\ &= -.0072 \end{aligned}$$

Improved bootstrap bias estimation: Efron & Tibshirani (1993), Section 10.4

Bias corrected estimates:

$$\begin{aligned} \bar{t}_n &= t_n - \widehat{Bias}_b(t_n) \\ &= 2t_n - \left(\frac{1}{B}\sum_{b=1}^{B} t^*_{n,b}\right) \end{aligned}$$

Law School example:

$$\begin{aligned} \tilde{r}_{15} &= r_{15} - \widehat{Bias}_b(r_{15}) \\ &= .7764 - (-.0072) = 0.7836 \end{aligned}$$

↑
here we moved farther away from $\rho = .7600$.

Bias correction can be dangerous in practice:

- $\bar{t}_n$ may have a substantially larger variance than t_n
- $MSE_F(\bar{t}_n)$
  $= [Bias_F(\bar{t}_n)]^2 + Var_F(\bar{t}_n)$
  is often larger than
  $MSE_F(t_n)$
  $= [Bias_F(t_n)]^2 + Var_F(t_n)$

Empirical Percentile Bootstrap Confidence Intervals

Construct an approximate
$(1-\alpha)\times 100\%$
confidence interval for θ.

- t_n is an estimator for θ
- Compute B bootstrap samples to obtain $t^*_{n,1}, \ldots, t^*_{n,B}$
- Order the bootstrapped values from smallest to largest

$$t_{n,(1)} \le t_{n,(2)} \le \ldots \le t_{n,(B)}$$

- Compute upper and lower $\frac{\alpha}{2} \times 100$th percentiles

  Compute

$$k_L = \left[(B+1)\frac{\alpha}{2}\right]$$

$$= \text{largest integer } \leq (B+1)\frac{\alpha}{2}$$

$$k_U = B + 1 - k_L$$

  Then, an approximate $(1-\alpha) \times 100\%$ confidence interval for θ is

$$[t_{n,(k_L)},\ t_{n,(k_U)}]$$

Law School example:

($B = 5000$ bootstrap samples)

Construct an approximate 90% confidence interval for

$\rho =$ population correlation

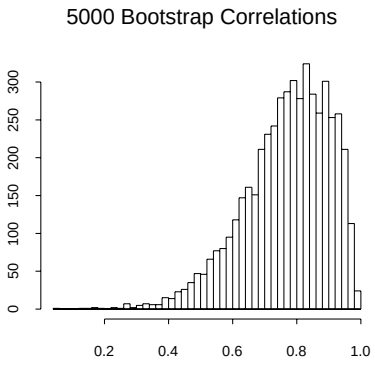
$$\alpha = .10$$

$$k_L = [(5001)(.05)] = [250.05] = 250$$

$$k_U = 5001 - 250 = 4751$$

An approximate large sample 90% confidence interval is

$$[r_{15,(250)}, r_{15,(4751)}] = [0.5207, 0.9487]$$



Law School example:

- Fisher Z-transformation

$$Z_n = \frac{1}{2}\log\left(\frac{1+r_n}{1-r_n}\right)$$

$$\dot{\sim}\ N\left(\frac{1}{2}\log\left(\frac{1+\rho}{1-\rho}\right), \frac{1}{n-3}\right)$$

- An approximate $(1-\alpha) \times 100\%$ confidence interval for $\frac{1}{2}\log\left(\frac{1+\rho}{1-\rho}\right)$ is

  lower limit: $Z_n - Z_{\alpha/2}\frac{1}{\sqrt{n-3}} = Z_L$

  upper limit: $Z_n + Z_{\alpha/2}\frac{1}{\sqrt{n-3}} = Z_U$

- Transform back to the original scale

$$\left[\frac{e^{2Z_L}-1}{e^{2Z_L}+1}, \frac{e^{2Z_u}-1}{e^{2Z_u}+1}\right]$$

For $n = 15$ and $r_{15} = .7764$ we have

$$Z_{15} = \frac{1}{2}\log\left(\frac{1+.7764}{1-.7764}\right) = 1.03624$$

and

$$Z_L = Z_{15} - Z_{.05}\frac{1}{\sqrt{15-3}} = .561372$$

$$Z_U = Z_{15} + Z_{.05}\frac{1}{\sqrt{15-3}} = 1.511113$$

and an approximate 90% confidence interval for the correlation is

$$(0.509, 0.907)$$

The bootstrap percentile interval would approximate this interval if the original sample was taken from a bivariate normal approximation.

Percentile Bootstrap Confidence Intervals

Suppose there is a transformation

$$\phi = m(\theta)$$

such that

$$\hat{\phi} = m(t_n) \sim N(\phi, \omega^2)$$

for some standard deviation ω. Then, an approximate confidence interval for θ is

$$[m^{-1}(\hat{\phi} - Z_{(\alpha/2)}\omega), m^{-1}(\hat{\phi} + Z_{\alpha/2}\omega)]$$

- The bootstrap percentile interval is a consistent approximation. (You do not have to identify the m() transformation.
- The bootstrap approximation becomes more accurate for larger sample
- For smaller samples, the coverage probability of the bootstrap percentile interval tends to be smaller than the nominal level $(1-\alpha)\times 100$.

- A percentile interval is entirely inside the parameter space.

  A percentile confidence interval for a correlation lies inside the interval $[-1, 1]$.

Bias-corrected and accelerated (BC_a) bootstrap percentile confidence intervals

- simulate B bootstrap samples and order the resulting estimates

$$t^*_{n,(1)} \leq t^*_{n,(2)} \leq \cdots \leq t^*_{n,(B)}$$

- The BC_a interval of intended coverage $1-\alpha$ is given by

$$[t^*_{n,([\alpha_1(B+1)])}, t^*_{n,([\alpha_2(B+1)])}]$$

where

$$\alpha_1 = \Phi\left(\bar{Z}_0 + \frac{\bar{Z}_0 - Z_{\alpha/2}}{1 - \bar{a}(\bar{Z}_0 - Z_{\alpha/2})}\right)$$

$$\alpha_2 = \Phi\left(\bar{Z}_0 + \frac{\bar{Z}_0 + Z_{\alpha/2}}{1 - \bar{a}(\bar{Z}_0 + Z_{\alpha/2})}\right)$$

and

$\Phi()$ is the c.d.f. for the standard normal distribution

$Z_{\alpha/2}$ is an "upper" percentile of the standard normal distribution, e.g., $Z_{.05} = 1.645$ and $\Phi(Z_{\alpha/2}) = 1 - \frac{\alpha}{2}$

$\bar{Z}_0 = \Phi^{-1}($proportion of $t^*_{n,b}$ values smaller than $t_n)$

is roughly a measure of *median bias* of t_n in "normal units."

When exactly half of the bootstrap samples have $t^*_{n,b}$ values less than t_n, then $\bar{Z}_0 = 0$.

$$\bar{a} = \frac{\sum_{j=1}^n (t_{n,(\cdot)} - t_{n,-j})^3}{6\left[\sum_{j=1}^n (t_{n,(\cdot)} - t_{n,-j})^2\right]^{3/2}}$$

is the estimated acceleration, where

$t_{n,-j}$ is the value of t_n when the j-th case is removed from the sample

and

$$t_{n,(\cdot)} = \frac{1}{n}\sum_{j=1}^n (t_{n,-j})$$

- BC_a intervals are second order accurate

  $Pr\{\theta <$ lower end of BC_a interval$\}$

  $$= \frac{\alpha}{2} + \frac{C_{lower}}{n}$$

  $Pr\{\theta >$ upper end of BC_a interval$\}$

  $$= \frac{\alpha}{2} + \frac{C_{upper}}{n}$$

- Bootstrap percentile intervals are first order accurate

  $$Pr\{\theta < \text{lower end}\} = \frac{\alpha}{2} + \frac{C^*_{lower}}{\sqrt{n}}$$

  $$Pr\{\theta < \text{upper end}\} = \frac{\alpha}{2} + \frac{C^*_{upper}}{\sqrt{n}}$$

- ABC intervals are approximations to BC_a intervals
  - second order accurate
  - only use about 3% of the computation time

  (See Efron & Tibshirani (1993) Chapter 14)


```
#  This is S-plus code for creating
#  bootstrapped confidence intervals
#  for a correlation coefficient.  It is
#  stored in the file
#
#              lawschl.ssc
#
#  Any line preceded with a pound sign
#  is a comment that is ignored by the
#  program.  The law school data are
#  read from the file
#
#              lawschl.dat

#  Enter the law school data into a data frame


  laws <- read.table("lawschl.dat",
            col.names=c("School","LSAT","GPA"))
  laws
```



```
   School LSAT  GPA
1       1  576 3.39
2       2  635 3.30
3       3  558 2.81
4       4  578 3.03
5       5  666 3.44
6       6  580 3.07
7       7  555 3.00
8       8  661 3.43
9       9  651 3.36
10     10  605 3.13
11     11  653 3.12
12     12  575 2.74
13     13  545 2.76
14     14  572 2.88
15     15  594 2.96
```

```
#  Plot the data

par(fin=c(7.0,7.0),pch=16,mkh=.15,mex=1.5)
plot(laws$LSAT,laws$GPA, type="p",
         xlab="GPA",ylab="LSAT score",
         main="Law School Data")


#  Compute the sample correlation matrix

  rr<-cor(laws$LSAT,laws$GPA)
  cat("Estimated correlation:  ",
                  round(rr,5), fill=T)


Estimated correlation:   0.77637
```



```
# First test for zero correlation

   n <- length(laws$LSAT);
   tt<- sqrt(n-2)*rr/sqrt(1 - rr*rr)
   pval <- 1 - pt(tt,n-2)
   pval <- round(pval,digits=5)

   cat("t-test for zero correlation: ",
                round(tt,4), fill=T)

t-test for zero correlation:  4.4413

  cat("p-values for the t-test for
    zero correlation: ", pval, fill=T)

p-values for the t-test
   for zero correlation:  0.00033
```



```
# Use Fisher's z-transformation to construct
# approximate confidence intervals.
# First set the level of confidence at 1-alpha.

   alpha <- .10
   z <- 0.5*log((1+rr)/(1-rr))
   zl <- z - qnorm(1-alpha/2)/sqrt(n-3)
   zu <- z + qnorm(1-alpha/2)/sqrt(n-3)
   rl <- round((exp(2*zl)-1)/(exp(2*zl)+1),
               digits=4)
   ru <- round((exp(2*zu)-1)/(exp(2*zu)+1),
               digits=4)

   per <- (1-alpha)*100;
   cat( per,"% confidence interval:  (",
                  rl,", ",ru,")",fill=T)


90 % confidence interval:  ( 0.509 ,  0.9071 )
```



```
# Compute bootstrap confidence intervals.
# Use B=5000 bootstrap samples.

  nboot <- 5000
  rboot <- bootstrap(data=laws,
           statistic=cor(GPA,LSAT),B=nboot)

Forming replications  1  to  100
Forming replications  101  to  200
Forming replications  201  to  300
     .
     .
     .
Forming replications  4801  to  4900
Forming replications  4901  to  5000
```

```
# limits.emp(): Calculates empirical percentiles
#               for the bootstrapped parameter
#               estimates in a resamp object.
#               The quantile function is used to
#             calculate the empirical percentiles.
# usage:
# limits.emp(x, probs=c(0.025, 0.05, 0.95, 0.975))

  limits.emp(rboot, probs=c(0.05,0.95))

            5%       95%
Param 0.523511 0.9473424


#  Do another set of 5000 bootstrapped values

  rboot <- bootstrap(data=laws,
               statistic=cor(GPA,LSAT),B=nboot)
  limits.emp(rboot, probs=c(0.05,0.95))


            5%       95%
Param 0.520661 0.9486835
```



```
# limits.bca():  Calculates BCa (bootstrap
#                bias-correct, adjusted)
#                confidence limits.
# usage:
#     limits.bca(boot.obj,
#        probs=c(0.025, 0.05, 0.95, 0.975),
#            details=F, z0=NULL,
#            acceleration=NULL,
#            group.size=NULL,
#            frame.eval.jack=sys.parent(1))

  limits.bca(rboot,probs=c(0.05,0.95),detail=T)

$limits:
            5%       95%
Param 0.442216 0.9306627

$emp.probs:
              5%       95%
Param 0.01954177 0.9065055
```

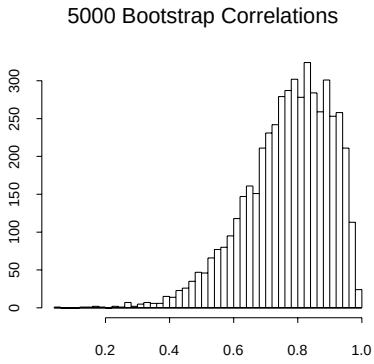


```
$z0:
       Param
 -0.07979538

$acceleration:
       Param
 -0.07567156

$group.size:
[1] 1
```



```
#   Both sets of confidence intervals could
#   have been obtained from the summary(  )
#   function

  summary(rboot)


Call:
bootstrap(data = laws, statistic =
             cor(GPA, LSAT), B = nboot)

Number of Replications: 5000

Summary Statistics:
      Observed      Bias   Mean     SE
Param   0.7764 -0.007183 0.7692 0.1341

Empirical Percentiles:
        2.5%     5%    95%  97.5%
Param 0.4638 0.5207 0.9487 0.9616

BCa Percentiles:
        2.5%     5%    95%  97.5%
Param 0.3462 0.4422 0.9307 0.9449
```

```
# Make a histogram of the bootstrapped
# correlations.

  hist(rboot$rep,nclass=50,
           xlab=" ",
           main="5000 Bootstrap Correlations",
           density=.0001)
```

The bootstrap can fail:

Example: $X_1, X_2, \ldots, X_n$ are sampled from a uniform $(0, \theta)$ distribution:

true density:

$$f(x) = \frac{1}{\theta}, \qquad 0 < x < \theta$$

true c.d.f.: $F(x) = \begin{cases} 0 & x \leq 0 \\ \frac{x}{\theta} & 0 < x \leq \theta \\ 1 & x > \theta \end{cases}$

Bootstrap percentile confidence intervals for θ tend to be too short.

Application of the bootstrap must adequately replicate the random process that produced the original sample

- Simple random samples
- "Nested" experiments
  - Sample plants from a field
  - Sample leaves from plants
- Curve fitting (existence of covariates)
  - Fixed levels
  - Random samples

Parametric Bootstrap

- Suppose you "knew" that

$$(X_{1j}, X_{2j}) \qquad j = 1, \ldots, n$$

were obtained from a simple random sample from a bivariate normal distribution, i.e.,

$$\mathrm{X}_j = \begin{bmatrix} X_{1j} \\ X_{2j} \end{bmatrix} \sim NID\left(\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \Sigma \right)$$

- Estimate unknown parameters

$$\bar{\mu} = \begin{bmatrix} \bar{\mu}_1 \\ \bar{\mu}_2 \end{bmatrix} = \frac{1}{n} \sum_{j=1}^{n} \mathrm{X}_j = \overline{\mathrm{X}}$$

$$\Sigma = \frac{1}{n-1} \sum_{j=1}^{n} (\mathrm{X}_j - \overline{\mathrm{X}})(\mathrm{X}_j - \overline{\mathrm{X}})^T$$

- Obtain a bootstrap sample of size n by sampling from a $N(\hat{\mu}, \hat{\Sigma})$ distribution, i.e.

$$\mathrm{X}_{1,b}, \ldots, \mathrm{X}_{n,b}$$

and compute $t^*_{n,b} = r_{n,b}$

- Repeat this to obtain

$$r_{n,1}, r_{n,2}, \ldots, r_{n,B}$$

- Compute bootstrap
  - standard errors
  - bias estimators
  - confidence intervals

References

Davison, A.C. and Hinkley, D.V. (1997) Bootstrap Methods and Their Applications, Cambridge Series in Statistical and Probabilistic Mathematics 1, Cambridge University Press, New York.

Efron, B. (1982) The Jackknife, The Bootstrap and other resampling plans, CBMS, 38, SIAM-NSF, Philadelphia.

Efron, B. and Gong, G. (1983) The American Statistician, , 36-48.

Efron, B. (1987) Better Bootstrap confidence intervals (with discussion) Journal of the American Statistical Association, 82, 171-200.

Efron, B. and Tibsharani, R. (1993) An Introduction to the Bootstrap, Chapman and Hall, New York.

Shao, J. and Tu, D. (1995) The Jackknife and Bootstrap, New York, Springer.

Example 12.2: Stormer viscometer data *(Venables & Ripley, Chapter 8)*

- measure viscosity of a fluid
- measure time taken for an inner cyclinder in the mechanism to complete a specific number of revolutions in response to an actuating weight
- calibrate the viscometer using runs with
  - varying weights (W) (g)
  - fluids with known viscosity (V)
  - record the time (T) (sec

- theoretical model

$$T = \frac{\beta_1 V}{W - \beta_2} + \epsilon$$

```
#  This code is used to explore the
#  Stormer viscometer data.  It is stored
#  in the file
#
#            stormer.ssc
#
#  Enter the data into a data frame.
#  The data are stored in the file
#
#            stormer.dat

library(MASS)

stormer <- read.table("stormer.dat")
stormer
```



```
   Viscosity  Wt  Time
1       14.7  20  35.6
2       27.5  20  54.3
3       42.0  20  75.6
4       75.7  20 121.2
5       89.7  20 150.8
6      146.6  20 229.0
7      158.3  20 270.0
8       14.7  50  17.6
9       27.5  50  24.3
10      42.0  50  31.4
11      75.7  50  47.2
12      89.7  50  58.3
13     146.6  50  85.6
14     158.3  50 101.1
15     161.1  50  92.2
16     298.3  50 187.2
17      75.7 100  24.6
18      89.7 100  30.0
19     146.6 100  41.7
20     158.3 100  50.3
21     161.1 100  45.1
22     298.3 100  89.0
23     298.3 100  86.5
```


Starting values for β_1 and β_2:

Fit an approximate linear model. Note that

$$T_i = \frac{\beta_1 V_i}{W_i - \beta_2} + \epsilon_i$$

$$\Rightarrow \quad (W_i - \beta_2) T_i = \beta_1 V_i + \epsilon_i (W_i - \beta_2)$$

$$\Rightarrow \quad W_i T_i = \beta_1 V_i + \beta_2 T_i + \epsilon_i (W_1 - \beta_2)$$

this is the new response variable

this is the new error

Use OLS estimation to obtain

$$\widehat{\beta}_1^{(0)} = 28.876$$

$$\widehat{\beta}_2^{(0)} = 2.8437$$


```
#  Use a linear approximation to obtain
#  starting values for least squares
#  estimation in the non-linear model

fm0 <- lm(Wt*Time ~ Viscosity + Time - 1,
                      data=stormer)
b0 <- coef(fm0)
names(b0) <- c("b1","b2")


#  Fit the non-linear model

storm.fm <- nls(
   formula = Time ~ b1*Viscosity/(Wt-b2),
      data = stormer,  start = b0,
     trace = T)


885.365 : 28.8755 2.84373
825.11  : 29.3935 2.23328
825.051 : 29.4013 2.21823
```

```
 summary(storm.fm)$parameters

       Value   Std. Error     t value
b1 29.401328    0.9155353   32.113813
b2  2.218226    0.6655234    3.333054
```



```
#  Create a bivariate confidence region
#  for the the (b1,b2) parameters.
#  First set up a grid of (b1,b2) values

bc <- coef(storm.fm)
se <- sqrt(diag(vcov(storm.fm)))
dv <- deviance(storm.fm)

gsize<-51
b1 <- bc[1] + seq(-3*se[1], 3*se[1],
                        length = gsize)
b2 <- bc[2] + seq(-3*se[2], 3*se[2],
                        length = gsize)
bv <- expand.grid(b1, b2)

#  Create a function to evaluate sums of squares

ssq <- function(b)
   sum((stormer$Time - b[1] * stormer$Viscosity/
               (stormer$Wt-b[2]))^2)
```



```
#  Evalute the sum of squared residuals and
#  approximate F-ratios for all of the
#  (b1,b2) values on the grid

dbeta <- apply(bv, 1, ssq)
n<-length(stormer$Viscosity)
df1<-length(bc)
df2<-n-df1
fstat <- matrix( ((dbeta - dv)/df1) / (dv/df2),
                        gsize, gsize)
```



```
#  Create the plot

par(fin=c(7.0,7.0), mex=1.5,lwd=3)
plot(b1, b2, type="n",
           main="95% Confidence Region")
contour(b1, b2, fstat, levels=c(1,2,5,7,10,15,20),
                  labex=0.75, lty=2, add=T)
contour(b1, b2, fstat, levels=qf(0.95,2,21),
                  labex=0, lty=1, add=T)
text(31.6,0.3,"95% CR", adj=0, cex=0.75)
points(bc[1], bc[2], pch=3, mkh=0.15)

# remove b1,b2, and bv

rm(b1,b2,bv,fstat)
```

Construct a joint confidence region for (β_1, β_2):

- Deviance (residual sum of squares):

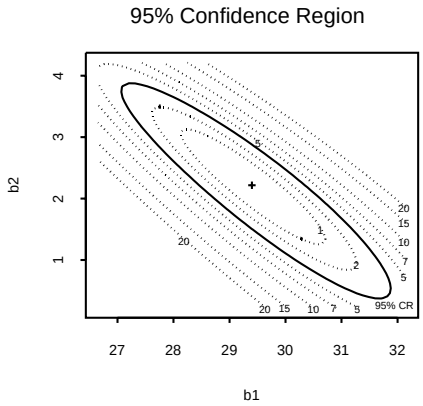
$$d(\beta_1, \beta_2) = \sum_{j=1}^{n} \left[T_j - \frac{\beta_1 V_j}{W_j - \beta_2} \right]^2$$

- Approximate F-statistic:

$$F(\beta_1, \beta_2) = \frac{\frac{d(\beta_1,\beta_2) - d(\hat{\beta}_1,\hat{\beta}_2)}{2}}{\frac{d(\beta_1,\beta_2)}{n-2}}$$

- An approximate $(1-\alpha) \times 100\%$ confidence interval consists of all (β_1, β_2) such that

$$F(\beta_1, \beta_2) < F_{(2,n-2),\alpha}$$



Bootstrap Estimation

Bootstrap I:

Sample $n = 23$ cases

$$(T_i, W_i, V_i)$$

from the original data set (using simple random sampling with replacement)


```
#=======================================
# Bootstrap I:
#      Treat the regressors as random and
#      resample the cases (y,x_1,x_2)
#=======================================

storm.boot1 <- bootstrap(stormer,
      coef(nls(Time~b1*Viscosity/(Wt-b2),
      data=stormer,start=bc)),B=1000)


summary(storm.boot1)

Call:
bootstrap(data = stormer,
      statistic = coef(nls(Time ~ (b1 *
Viscosity)/(Wt - b2),
      data = stormer, start = bc)), B = 1000)

Number of Replications: 1000
```

```
Summary Statistics:
   Observed     Bias   Mean     SE
b1   29.401 -0.08665 29.315 0.7194
b2    2.218  0.09412  2.312 0.8424

Empirical Percentiles:
      2.5%     5%    95%  97.5%
b1 27.8202 28.113 30.402 30.613
b2  0.8453  1.057  3.513  3.779

BCa Percentiles:
      2.5%     5%    95%  97.5%
b1 27.9368 28.242 30.525 30.747
b2  0.7462  0.899  3.409  3.604

Correlation of Replicates:
        b1      b2
b1  1.0000 -0.8628
b2 -0.8628  1.0000
```



```
#  Produce histograms of the bootstrapped
#  values of  the regression coefficients

# The following code is used to draw
# non-parametric estimated densities,
# normal densities, and a histogram
# on the same graph.

# truehist(): Plot a Histogram (prob=T
#             by default) For the function
#             hist(), probability=F by
#             default.
# width.SJ(): Bandwidth Selection by Pilot
#             Estimation of Derivatives.
#             Uses the method of Sheather
#             & Jones (1991) to  select  the
#             bandwidth of a Gaussian kernel
#             density estimator.
```



```
# density() : Estimate Probability Density
#             Function.  Returns x and y
#             coordinates of a non-parametric
#             estimate of the probability
#             density of the data.  Options
#             include the type of window to
#             use and the number of points
#             at which to estimate the density.
#             n = the number of equally spaced
#                 points at which the density
#                 is evaluated.
```

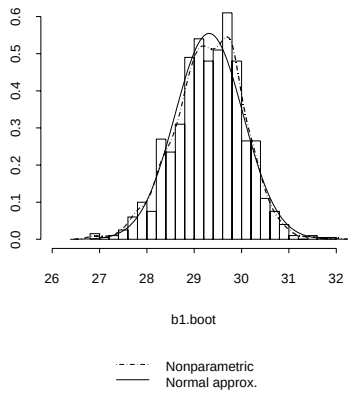


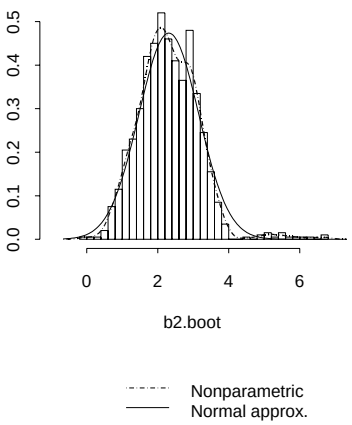
```
  b1.boot <- storm.boot1$rep[,1]
  b2.boot <- storm.boot1$rep[,2]

 library(MASS)

 par(fin=c(7.0,7.0),mex=1.5)

 mm <- range(b1.boot)
 min.int <- floor(mm[1])
 max.int <- ceiling(mm[2])

 truehist(b1.boot,xlim=c(min.int,max.int),
             density=.0001)

 width.SJ(b1.boot)

[1] 0.6918326


 b1.boot.dns1 <- density(b1.boot,n=200,width=.6)
 b1.boot.dns2 <-
    list( x = b1.boot.dns1$x,
  y = dnorm(b1.boot.dns1$x,mean(b1.boot),
              sqrt(var(b1.boot))))
```

```
 # Draw the non-parametric density

  lines(b1.boot.dns1,lty=3,lwd=2)

 # Draw normal densities

  lines(b1.boot.dns2,lty=1,lwd=2)

  legend(27.,-0.30,c("Nonparametric",
         "Normal approx."),
         lty=c(3,1),bty="n",lwd=2)


#  Display the distribution of the estimate
#  of the second parameter

  par(fin=c(7.0,7.0),mex=1.5)

  mm <- range(b2.boot)
  min.int <- floor(mm[1])
  max.int <- ceiling(mm[2])
```



```
  truehist(b2.boot,xlim=c(min.int,max.int),
              density=.001)

  width.SJ(b2.boot)
[1] 0.7339957


  b2.boot.dns1 <- density(b2.boot,n=200,width=.8)
  b2.boot.dns2 <-
     list( x = b2.boot.dns1$x,
   y = dnorm(b2.boot.dns1$x,mean(b2.boot),
               sqrt(var(b2.boot))))

  lines(b2.boot.dns1,lty=3,lwd=2)
  lines(b2.boot.dns2,lty=1,lwd=2)
  legend(1.,-0.30,c("Nonparametric",
          "Normal approx."),
          lty=c(3,1),bty="n",lwd=2)
```



Bootstrap II: Fix the values of the explanatory variables

$$\{(W_j, V_j) : j = 1, \ldots, n\}$$

- Compute residuals from fitting the model to the original sample

$$e_j = T_j - \frac{\hat{\beta}_1 V_j}{W_j - \hat{\beta}_2} \quad j = 1, \ldots, n$$

- Approximate sampling from the population of random errors by taking a sample (with replacement) from $\{e_1, \ldots, e_n\}$ say,

$$e^*_{1,b},\ e^*_{2,b}, \ldots,\ e^*_{n,b}$$

- Create new observations:

$$(T^*_{j,b}, W_j, V_j) \quad j = 1, \ldots, n$$

where

$$T^*_{j,b} = \frac{\hat{\beta}_1 V_j}{W_j - \hat{\beta}_2} + e^*_j$$

- Fit the model to the j-th bootstrap sample to obtain

$$\hat{\beta}^*_{1,b} \quad \text{and} \quad \hat{\beta}^*_{2,b}$$

- Repeat this for $b = 1, \ldots, B$ bootstrap samples


```
#=====================================
# Bootstrap II:
#      Treat the regressors as fixed and
#      resample from the residuals
#=====================================

# Center the residuals at zero and
# divide residuals by linear approximations
# to a multiple of the standard errors.  These
# centered and scaled residuals approximately
# have the same first two moments as the
# random errors, but they are not quite
# uncorrelated.

rs <- scale(resid(storm.fm),center=T,scale=F)

grad.f <- deriv3( expr = Y ~ (b1*X1)/(X2-b2),
              namevec =  c("b1", "b2"),
 function.arg = function(b1, b2, X1, X2, Y) NULL)
```



```
g2 <- grad.f(b[1], b[2], stormer$Viscosity,
          stormer$Wt, stormer$Tim)
D <- attr(g2, "gradient")
h <- 1-diag(D%*%solve(t(D)%*%D)%*%t(D))
rs <- rs/sqrt(h)
dfe <- length(rs)-length(coef(storm.fm))
vr <- var(rs)
rs <- rs%*%sqrt(deviance(storm.fm)/dfe/vr)
```

```
#  Create a function to use in fitting
#  the model to bootstrap samples

  storm.bf2 <- function(rs)
    {assign("Tim", fitted(storm.fm) + rs,
          frame = 1)
    nls(formula =  Tim ~ (b1*Viscosity)/(Wt-b2),
           data = stormer,
          start = coef(storm.fm)
           )$parameters }


 summary(storm.fm)$parameters

       Value   Std. Error     t value
b1 29.401328    0.9155353   32.113813
b2  2.218226    0.6655234    3.333054
```



```
#  Compute 1000 bootstrapped values of the
#  regression parameters

storm.boot2 <- bootstrap(data=rs,
     statistic=storm.bf2(rs),B=1000)

Forming replications  1  to  100
Forming replications  101  to  200
Forming replications  201  to  300
Forming replications  301  to  400
Forming replications  401  to  500
Forming replications  501  to  600
Forming replications  601  to  700
Forming replications  701  to  800
Forming replications  801  to  900
Forming replications  901  to  1000

Call:
bootstrap(data = rs, statistic = storm.bf2(rs),
            B = 1000)

b1.boot <- storm.boot2$rep[,1]
b2.boot <- storm.boot2$rep[,2]
```



```
#  The BCA intervals may not be correctly
#  computed by the following function

summary(storm.boot2)

Number of Replications: 1000

Summary Statistics:
   Observed    Bias   Mean     SE
b1   28.745  0.5719 29.317 0.8892
b2    2.432 -0.1682  2.264 0.6353
```



```
Empirical Percentiles:
    2.5%     5%    95%  97.5%
b1 27.58 27.847 30.690 30.924
b2  1.03  1.241  3.295  3.459

BCa Confidence Limits:
     2.5%     5%    95%  97.5%
b1 26.236 26.267 29.698 29.985
b2  1.316  1.524  3.563  3.683

Correlation of Replicates:
        b1      b2
b1  1.0000 -0.9194
```

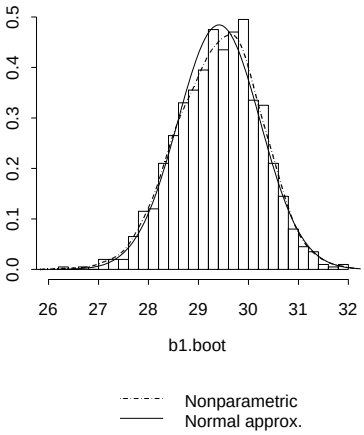
```
# The following code is used to draw
# non-parametric estimated densities,
# normal densities, and histograms
# on the same graphic window.

  par(fin=c(7.0,7.0),mex=1.3)

  mm <- range(b1.boot)
  min.int <- floor(mm[1])
  max.int <- ceiling(mm[2])

  truehist(b1.boot,xlim=c(min.int,max.int),
               density=.0001)

  b1.boot.dns1 <- density(b1.boot,n=200,
                   width=width.SJ(b1.boot))
  b1.boot.dns2 <-
     list( x = b1.boot.dns1$x,
   y = dnorm(b1.boot.dns1$x,mean(b1.boot),
              sqrt(var(b1.boot))))
```

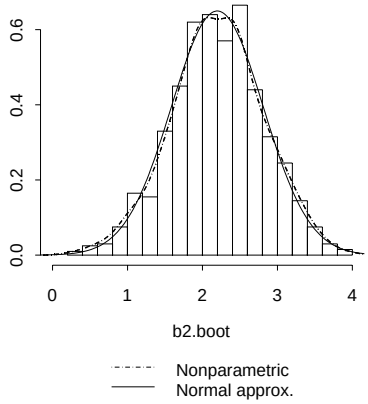


```
# Draw a non-parametric density

 lines(b1.boot.dns1,lty=3,lwd=2)

# draw normal densities

 lines(b1.boot.dns2,lty=1,lwd=2)

 legend(27,-0.22,c("Nonparametric",
               "Normal approx."),
            lty=c(3,1),bty="n",lwd=2)
```



```
#  Display the distribution for the other
#  parameter estimate


  par(fin=c(7.0,7.0),mex=1.3)

  mm <- range(b2.boot)
  min.int <- floor(mm[1])
  max.int <- ceiling(mm[2])

  truehist(b2.boot,xlim=c(min.int,max.int),
                density=.00001)

  b2.boot.dns1 <- density(b2.boot,n=200,
                width=width.SJ(b2.boot))
  b2.boot.dns2 <-
     list( x = b2.boot.dns1$x,
     y = dnorm(b2.boot.dns1$x,mean(b2.boot),
              sqrt(var(b2.boot))))

  lines(b2.boot.dns1,lty=3,lwd=2)
  lines(b2.boot.dns2,lty=1lwd=2)
  legend(0.5,-0.25,c("Nonparametric",
      "Normal approx."),lty=c(3,1),
       bty="n",lwd=2)
```



Comparison of standard errors:

Parameter	Asymptotic normal approx.	Random Bootstrap	Fixed Bootstrap
β_1	0.916	0.719	0.889
β_2	0.666	0.842	0.635

Comparison of approximate 95% confidence intervals:

Parameter	Asymptotic normal approx.	Random Bootstrap	Fixed Bootstrap
β_1	(27.50, 31.31)	(28.13, 30.40)	(27.42, 31.03)
β_2	(0.83, 3.60)	(1.06, 3.51)	(1.03, 3.46)

$$\uparrow$$

$$\widehat{\beta}_i \pm t_{21,.025} S_{\widehat{\beta}_i}$$