

Chapter 11

Introduction to the Bootstrap

11.1 Introduction

Statistical theory attempts to answer three basic questions:

1. How should I collect my data?
2. How should I analyze and summarize the data that I've collected?
3. How accurate are my data summaries?

Question 3 constitutes part of the process known as statistical inference. The bootstrap makes certain kinds of statistical inference. (Efron (1979), "Bootstrap methods: another look at the jackknife." Ann. Statist. 7, 1–26) Let's look at an example.

Example: Aspirin and heart attacks, large-sample theory Does aspirin prevent heart attacks in healthy middle-aged men? A controlled, randomized, double-blind study was conducted and gathered the following data.

	heart attacks (fatal plus non-fatal)	subjects
aspirin group:	104	11037
placebo group:	189	11034

A good experimental design, such as this one, simplifies the results! The ratio of the two rates (the risk ratio) is

$$\hat{\theta} = \frac{104/11037}{189/11034} = 0.55.$$

Because of the solid experimental design, we can believe that the aspirin-takers only have 55% as many heart attacks as the placebo-takers.

We are not really interested in the estimated ratio $\hat{\theta}$, but the true ratio, θ . That is the ratio if we could treat all possible subjects, not just a sample of them. Large sample theory tells us that the log risk ratio has an approximate Normal distribution. The standard error of the log risk ratio is estimated simply by the square root of the sum of the reciprocals of the four frequencies:

$$\text{SE}(\log(RR)) = \sqrt{\frac{1}{104} + \frac{1}{189} + \frac{1}{11037} + \frac{1}{11034}} = 0.1228$$

The 95% CI for $\log(\theta)$ is

$$\log(\hat{\theta}) \pm 1.96 \times \text{SE}(\log(RR)), \quad (-0.839, -0.357),$$

and exponentiating gives the CI on the ratio scale,

$$\exp\{\log(\hat{\theta}) \pm 1.96 \times \text{SE}(\log(RR))\}, \quad (0.432, 0.700).$$

The same data that allowed us to estimate the ratio θ with $\hat{\theta} = 0.55$ also allowed us to get an idea of the estimate's accuracy.

Example: Aspirin and strokes, large-sample theory The aspirin study tracked strokes as well as heart attacks.

	strokes	subjects
aspirin group:	119	11037
placebo group:	98	11034

The ratio of the two rates (the risk ratio) is

$$\hat{\theta} = \frac{119/11037}{98/11034} = 1.21.$$

It looks like aspirin is actually harmful, now, however the 95% interval for the true stroke ratio θ is (0.925, 1.583). This includes the neutral value $\theta = 1$, at which aspirin would be no better or worse than placebo for strokes.

11.2 Bootstrap

The bootstrap is a data-based simulation method for statistical inference, which can be used to produce inferences like those in the previous slides. The term “bootstrap” comes from literature. In “The Adventures of Baron Munchausen”, by Rudolph Erich Raspe, the Baron had fallen to the bottom of a deep lake, and he thought to get out by *pulling himself up by his own bootstraps*.

11.2.1 Ideal versus Bootstrap world, sampling distributions

Ideal world

1. Population of interest
2. Obtain many simple random samples (SRSs) of size n
3. For each SRS, calculate statistic of interest (θ)
4. Sampling distribution is the distribution of the calculated statistic

Bootstrap world

1. Population of interest; One empirical distribution based on a sample of size n

2. Obtain many bootstrap resamples of size n
3. For each resample, calculate statistic of interest (θ^*)
4. Bootstrap distribution is the distribution of the calculated statistic
5. Bootstrap distribution estimates the sampling distribution centered at the statistic (not the parameter).

Example: Aspirin and strokes, bootstrap Here's how the bootstrap works in the stroke example. We create two populations:

- the first consisting of 119 ones and $11037 - 119 = 10918$ zeros,
- the second consisting of 98 ones and $11034 - 98 = 10936$ zeros.

We draw with replacement a sample of 11037 items from the first population, and a sample of 11034 items from the second population. Each is called a *bootstrap sample*. From these we derive the bootstrap replicate of $\hat{\theta}$:

$$\hat{\theta}^* = \frac{\text{Proportion of ones in bootstrap sample 1}}{\text{Proportion of ones in bootstrap sample 2}}$$

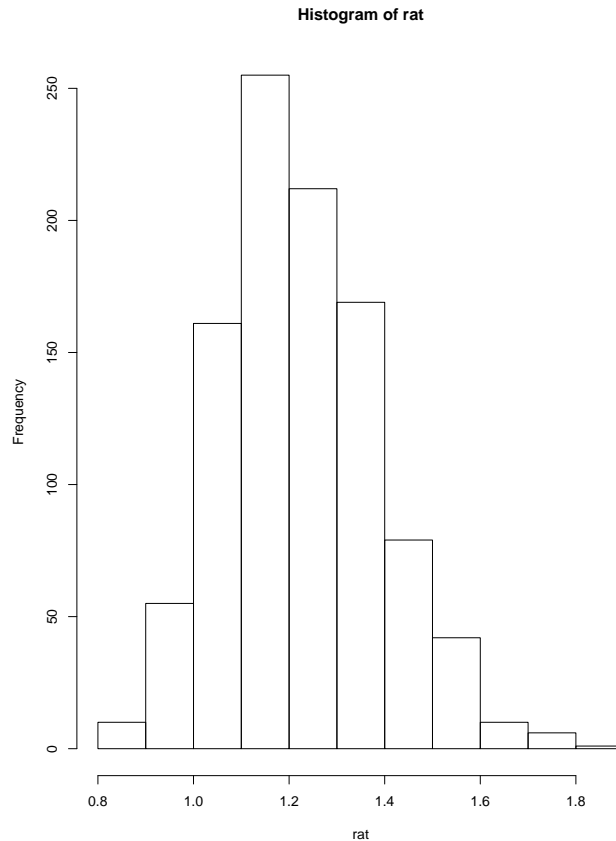
Repeat this process a large number of times, say 1000 times, and obtain 1000 *bootstrap replicates* $\hat{\theta}^*$. The summaries are in the code, followed by a histogram of bootstrap replicates, $\hat{\theta}^*$.

```
# Implementation in R
set.seed(8); # random number seed for sampling repeatability

n1=11037; s1=119; # sample size and number of successes sample 1
n2=11034; s2=98; # sample size and number of successes sample 2
p1 = c(rep(1,s1), rep(0,n1-s1)); # sample 1
p2 = c(rep(1,s2), rep(0,n2-s2)); # sample 2

n.bs=1000; # draw n.bs bootstrap samples
bs1=rep(0,n.bs); bs2=rep(0,n.bs); # init location for bootstrap samples
for (i in 1:n.bs) {
  # proportion of successes in bootstrap samples 1 and 2
  bs1[i] = sum(sample(p1, n1, replace = TRUE))/n1;
  bs2[i] = sum(sample(p2, n2, replace = TRUE))/n2;
}

rat = bs1/bs2; # bootstrap replicates of ratio estimates
hist(rat); # histogram of ratio estimates
rats = sort(rat); # sort the ratio estimates to obtain bootstrap CI
# 25th and 976th sorted values are the bootstrap CI
CI.bs = c(rats[round(0.025*n.bs)], rats[round(0.976*n.bs)]); CI.bs
# [1] 0.9343254 1.5760421
```



In this simple case, the confidence interval derived from the bootstrap (0.934, 1.576) agrees very closely with the one derived from statistical theory (0.925, 1.583). Bootstrap methods are intended to simplify the calculation of inferences like those using large-sample theory, producing them in an automatic way even in situations much more complicated than the risk ratio in the aspirin example.

11.2.2 The accuracy of the sample mean

For sample means, and essentially *only* for sample means, an accuracy formula (for the standard error of the parameter) is easy to obtain (using the delta method). We'll see how to use the bootstrap for the sample mean, then for the more complicated situation of assessing the accuracy of the median.

Bootstrap Principle The **plug-in principle** is used when the underlying distribution is unknown and you substitute your best guess for what that distribution is. What to substitute?

- Empirical distribution — ordinary bootstrap
- Smoothed distribution — (kernel) smoothed bootstrap
- Parametric distribution — parametric bootstrap
- Satisfy assumptions, e.g. the null hypothesis

This substitution works in many cases, but not always. Keep in mind that the bootstrap distribution is centered at the statistic, not the parameter. Implementation is done by Monte Carlo sampling.

The bootstrap is commonly implemented in one of two ways, nonparametrically or parametrically. An *exact nonparametric bootstrap* requires n^n samples! That's one for every possible combination of each of n observation positions taking the value of each of n observations. This is sensibly approximated by using the Monte Carlo strategy of drawing a large number (1000 or 10000) of random resamples. On the other hand, a **parametric bootstrap** first assumes a distribution for the population (such as a normal distribution) and estimates the distributional parameters (such as the mean and variance) from the observed sample. Then, the Monte Carlo strategy is used to draw a large number (1000 or 10000) of samples from the estimated parametric distribution.

Example: Mouse survival, large-sample theory Sixteen mice were randomly assigned to a treatment group or a control group. Shown are their survival times, in days, following a test surgery. Did the treatment prolong survival?

Group	Data	(n)	Mean	est. SE
Treatment:	94, 197, 16, 38, 99, 141, 23	(7)	86.86	25.24
Control:	52, 104, 146, 10, 51, 30, 40, 27, 46	(9)	56.22	14.14
			Difference:	30.63 28.93

Stem plots of the data are below. There seems to be a slight difference in variability between the two treatment groups.

```
# Implementation in R
p1 = c(94, 197, 16, 38, 99, 141, 23); # sample 1
p2 = c(52, 104, 146, 10, 51, 30, 40, 27, 46); # sample 2
n1 = length(p1); n2 = length(p2); # sample sizes

stem(p1,scale=2);stem(p2,scale=2); # stem-and-leaf plots
```

The decimal point is 1 digit(s) to the right of the |

```
sample 1      sample 2
0 | 6          0 | 0
2 | 38        2 | 70
4 |           4 | 0612
6 |           6 |
8 | 49        8 |
10 |          10 | 4
12 |          12 |
14 | 1        14 | 6
16 |
18 | 7
```

The standard error for the difference is $28.93 = \sqrt{25.24^2 + 14.14^2}$, so the observed difference of 30.63 is only $30.63/28.93=1.05$ estimated standard errors greater than zero, an *insignificant* result.

The two-sample t -test of the difference in means confirms the lack of statistically significant difference between these two treatment groups with a p -value=0.3155.

```
# Implementation in R
# stack as a vector of values with vector of indicators
x=matrix(c(p1,p2,rep(1,n1),rep(2,n2)),ncol=2);
t.test(x[,1] ~ x[,2]); # perform two-sample t-test

Welch Two Sample t-test

data:  x[, 1] by x[, 2]
t = 1.0587, df = 9.654, p-value = 0.3155
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -34.15279  95.42263
sample estimates:
mean in group 1 mean in group 2
 86.85714      56.22222
```

But these are small samples, and certainly sample 2 does not look normal. We could do a nonparametric two-sample test of difference of medians. Or, we could use the bootstrap to make our inference.

Example: Mouse survival, two-sample bootstrap, mean Here's how the bootstrap works in the two-sample mouse example. We draw with replacement from each sample, calculate the mean for each sample, then take the difference in means. Each is called a *bootstrap sample* of the difference in means. From these we derive the bootstrap replicate of $\hat{\theta}$:

$$\hat{\theta}^* = \bar{x}^* - \bar{y}^*.$$

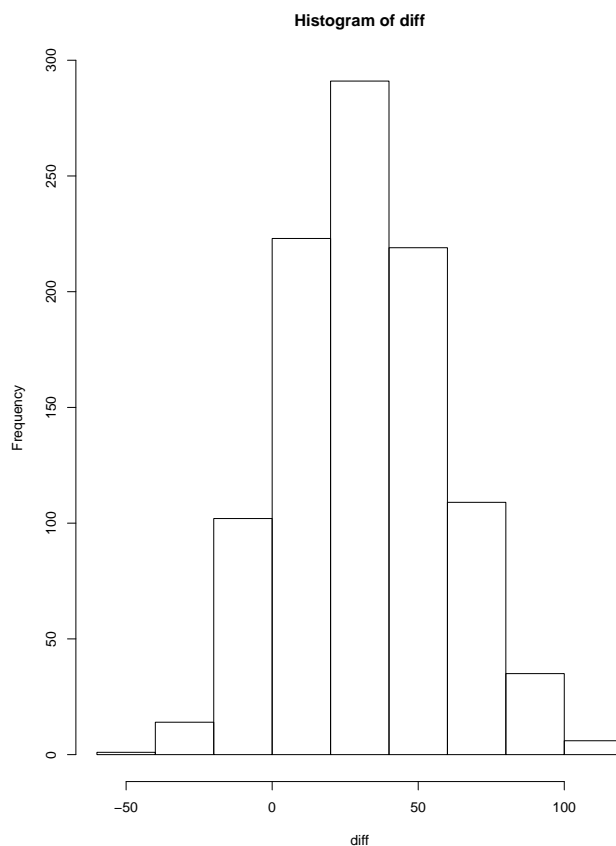
Repeat this process a large number of times, say 1000 times, and obtain 1000 *bootstrap replicates* $\hat{\theta}^*$. The summaries are in the code, followed by a histogram of bootstrap replicates, $\hat{\theta}^*$.

```
# Implementation in R
set.seed(8); # random number seed for sampling repeatability

n.bs=1000; # draw n.bs bootstrap samples
bs1=rep(0,n.bs); bs2=rep(0,n.bs); # init location for bootstrap samples
for (i in 1:n.bs) {
  bs1[i] = mean(sample(p1, n1, replace = TRUE)); # mean of
  bs2[i] = mean(sample(p2, n2, replace = TRUE)); # bootstrap samples 1 and 2
}

diff = bs1-bs2; # bootstrap replicates of difference estimates
hist(diff); # histogram of difference estimates
diffs = sort(diff); # sort the difference estimates to obtain bootstrap CI
# 25th and 976th sorted values are the bootstrap CI
CI.bs = c(diffs[round(0.025*n.bs)], diffs[round(0.976*n.bs)]); CI.bs
```

```
# [1] -17.33333 84.38095
sd(diff)
# [1] 26.28996
```



Example: Mouse survival, two-sample bootstrap, median For most statistics (such as the median) we don't have a formula for the limiting value of the standard error, but in fact no formula is needed. Instead, we use the numerical output of the bootstrap program. The summaries are in the code, followed by a histogram of bootstrap replicates, $\hat{\theta}^*$.

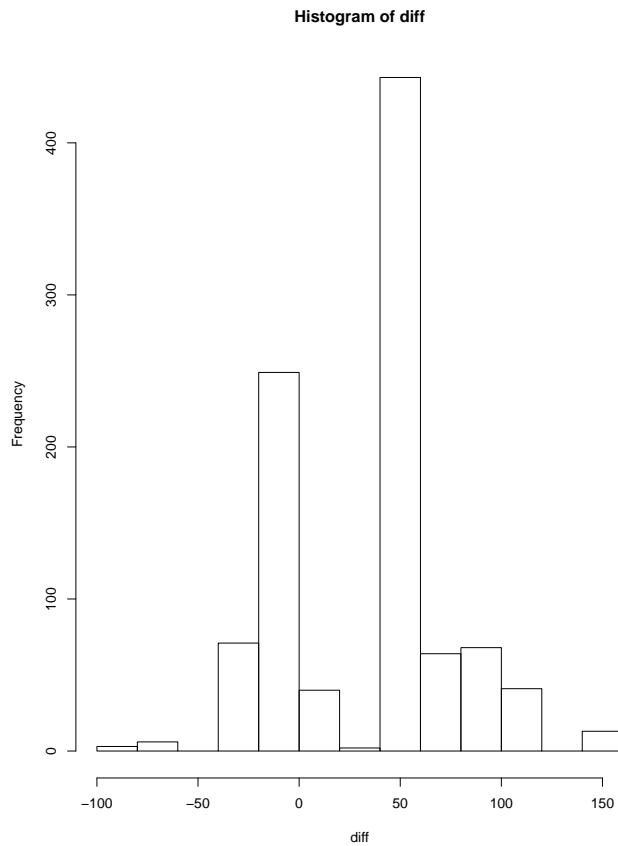
Group	Data	(n)	Median	est. SE
Treatment:	94, 197, 16, 38, 99, 141, 23	(7)	94	?
Control:	52, 104, 146, 10, 51, 30, 40, 27, 46	(9)	46	?
		Difference:	48	?

```
# Implementation in R
set.seed(8); # random number seed for sampling repeatability

p1 = c(94, 197, 16, 38, 99, 141, 23); # sample 1
p2 = c(52, 104, 146, 10, 51, 30, 40, 27, 46); # sample 2
n1 = length(p1); n2 = length(p2); # sample sizes
```

```
n.bs=1000; # draw n.bs bootstrap samples
bs1=rep(0,n.bs); bs2=rep(0,n.bs); # init location for bootstrap samples
for (i in 1:n.bs) {
  bs1[i] = median(sample(p1, n1, replace = TRUE)); # median of
  bs2[i] = median(sample(p2, n2, replace = TRUE)); # bootstrap samples 1 and 2
}

diff = bs1-bs2; # bootstrap replicates of difference estimates
hist(diff); # histogram of difference estimates
diffs = sort(diff); # sort the difference estimates to obtain bootstrap CI
# 25th and 976th sorted values are the bootstrap CI
CI.bs = c(diffs[round(0.025*n.bs)], diffs[round(0.976*n.bs)]); CI.bs
# [1] -29 111
sd(diff)
# [1] 40.79996
```



Example: Law School, correlation of (LSAT, GPA) The population of average student measurements of (LSAT, GPA) for the universe of 82 law schools are in the table below. Imagine that we don't have all 82 schools worth of data. Consider taking a random sample of 15 schools, indicated by the +'s.

School	LSAT	GPA	School	LSAT	GPA	School	LSAT	GPA
1	622	3.23	28	632	3.29	56	641	3.28
2	542	2.83	29	587	3.16	57	512	3.01
3	579	3.24	30	581	3.17	58	631	3.21
4+	653	3.12	31+	605	3.13	59	597	3.32
5	606	3.09	32	704	3.36	60	621	3.24
6+	576	3.39	33	477	2.57	61	617	3.03
7	620	3.10	34	591	3.02	62	637	3.33
8	615	3.40	35+	578	3.03	62	572	3.08
9	553	2.97	36+	572	2.88	64	610	3.13
10	607	2.91	37	615	3.37	65	562	3.01
11	558	3.11	38	606	3.20	66	635	3.30
12	596	3.24	39	603	3.23	67	614	3.15
13+	635	3.30	40	535	2.98	68	546	2.82
14	581	3.22	41	595	3.11	69	598	3.20
15+	661	3.43	42	575	2.92	70+	666	3.44
16	547	2.91	43	573	2.85	71	570	3.01
17	599	3.23	44	644	3.38	72	570	2.92
18	646	3.47	45+	545	2.76	73	605	3.45
19	622	3.15	46	645	3.27	74	565	3.15
20	611	3.33	47+	651	3.36	75	686	3.50
21	546	2.99	48	562	3.19	76	608	3.16
22	614	3.19	49	609	3.17	77	595	3.19
23	628	3.03	50+	555	3.00	78	590	3.15
24	575	3.01	51	586	3.11	79+	558	2.81
25	662	3.39	52+	580	3.07	80	611	3.16
26	627	3.41	53+	594	2.96	81	564	3.02
27	608	3.04	54	594	3.05	82+	575	2.74
			55	560	2.93			

Implementation in R

scan values from keyboard (allows copy/paste into window)

School LSAT GPA Sampled

law = scan()

```

1 622 3.23 0 2 542 2.83 0 3 579 3.24 0 4 653 3.12 1 5 606 3.09 0
6 576 3.39 1 7 620 3.10 0 8 615 3.40 0 9 553 2.97 0 10 607 2.91 0
11 558 3.11 0 12 596 3.24 0 13 635 3.30 1 14 581 3.22 0 15 661 3.43 1
16 547 2.91 0 17 599 3.23 0 18 646 3.47 0 19 622 3.15 0 20 611 3.33 0
21 546 2.99 0 22 614 3.19 0 23 628 3.03 0 24 575 3.01 0 25 662 3.39 0
26 627 3.41 0 27 608 3.04 0 28 632 3.29 0 29 587 3.16 0 30 581 3.17 0
31 605 3.13 1 32 704 3.36 0 33 477 2.57 0 34 591 3.02 0 35 578 3.03 1
36 572 2.88 1 37 615 3.37 0 38 606 3.20 0 39 603 3.23 0 40 535 2.98 0
41 595 3.11 0 42 575 2.92 0 43 573 2.85 0 44 644 3.38 0 45 545 2.76 1
46 645 3.27 0 47 651 3.36 1 48 562 3.19 0 49 609 3.17 0 50 555 3.00 1

```

```

51  586  3.11  0  52  580  3.07  1  53  594  2.96  1  54  594  3.05  0  55  560  2.93  0
56  641  3.28  0  57  512  3.01  0  58  631  3.21  0  59  597  3.32  0  60  621  3.24  0
61  617  3.03  0  62  637  3.33  0  62  572  3.08  0  64  610  3.13  0  65  562  3.01  0
66  635  3.30  0  67  614  3.15  0  68  546  2.82  0  69  598  3.20  0  70  666  3.44  1
71  570  3.01  0  72  570  2.92  0  73  605  3.45  0  74  565  3.15  0  75  686  3.50  0
76  608  3.16  0  77  595  3.19  0  78  590  3.15  0  79  558  2.81  1  80  611  3.16  0
81  564  3.02  0  82  575  2.74  1

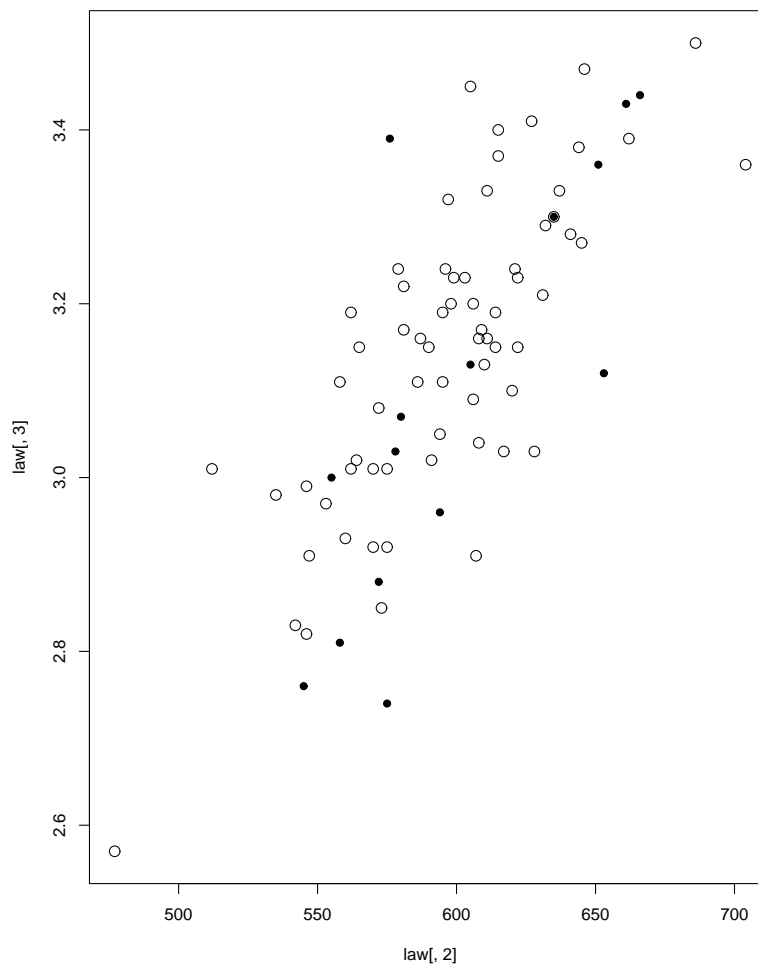
```

```

law = matrix(law,ncol=4,byrow=TRUE); # law matrix -- population
laws = matrix(law[(law[,4]==1)],ncol=4,byrow=FALSE); # sample
n1 = dim(law)[1]; n2 = dim(laws)[1]; # sample sizes
colnames(law) = c("School", "LSAT", "GPA", "Sampled"); # name the columns
plot(law[,2],law[,3], pch=1+19*law[,4], cex=1.4); # scatterplot

```

A scatterplot of LSAT (horizontal) and GPA (vertical) is below. The solid points indicate the 15 sampled schools.



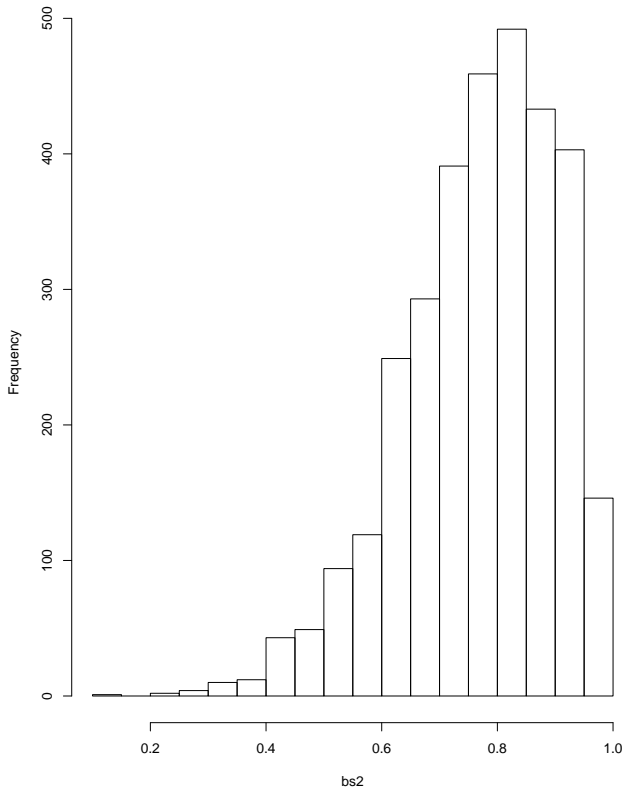
Let's bootstrap the sample of 15 observations to get the bootstrap sampling distribution of correlation (for sampling 15 from the population). From the bootstrap sampling distribution we'll

calculate a bootstrap confidence interval for the true population correlation, as well as a bootstrap standard deviation for the correlation. But how well does this work? Let's compare it against the *true* sampling distribution by drawing 15 random schools from the population of 82 schools and calculating the correlation. If the bootstrap works well (from our hopefully representative sample of 15), then the bootstrap sampling distribution from the 15 schools will be close to the true sampling distribution.

The code below does that, followed by two histograms. In this case, the histograms are noticeably non-normal, having a long tail toward the left. Inferences based on the normal curve are suspect when the bootstrap histogram is markedly non-normal. The histogram on the left is the nonparametric bootstrap sampling distribution using only the $n = 15$ sampled schools with 3200 bootstrap replicates of $\widehat{\text{corr}}(x^*)$. The histogram on the right is the true sampling distribution using 3200 replicates of $\widehat{\text{corr}}(x^*)$ from the population of law school data, repeatedly drawing $n = 15$ without replacement from the $N = 82$ points. Impressively, the bootstrap histogram on the left strongly resembles the population histogram on the right. Remember, in a real problem we would only have the information on the left, from which we would be trying to infer the situation on the right.

```
# Implementation in R
set.seed(8); # random number seed for sampling repeatability
n.bs=3200; # draw n.bs bootstrap samples
bs1=rep(0,n.bs); bs2=rep(0,n.bs); # init location for bootstrap samples
for (i in 1:n.bs) {
  # sample() draws indices then bootstrap correlation of LSAT and GPA
  bs1[i] = cor(law [sample(seq(n1), n2, replace = TRUE),2:3])[1,2]; # population
  bs2[i] = cor(laws[sample(seq(n2), n2, replace = TRUE),2:3])[1,2]; # sample
}
hist(bs1,nclass=20); # histogram of correlations for law -- population
hist(bs2,nclass=20); # histogram of correlations for laws -- sample
corrs = sort(bs1); # sort the difference estimates to obtain bootstrap CI
# 25th and 976th sorted values are the bootstrap CI
alpha=0.05;
CI.bs = c(corrs[round((alpha/2)*n.bs)], corrs[round((1-alpha/2)*n.bs)+1]); CI.bs
# [1] 0.4472529 0.9296237
sd(corrs)
# [1] 0.1279922
corrs = sort(bs2); # sort the difference estimates to obtain bootstrap CI
# 25th and 976th sorted values are the bootstrap CI
CI.bs = c(corrs[round((alpha/2)*n.bs)], corrs[round((1-alpha/2)*n.bs)+1]); CI.bs
# [1] 0.4617450 0.9586325
sd(corrs)
# [1] 0.1313620
```

Histogram of bs2



Histogram of bs1

