Chapter 1

Bootstrap

1.1 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 consitutes part of the process known as statistical inference. The bootstrap makes certain kinds of statistical inference[1]. 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.

[1]Efron (1979), "Bootstrap methods: another look at the jackknife." Ann. Statist. 7, 1–26

	(fatal plus non-fatal) heart attacks	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:

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

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

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

and expontiating gives the CI on the ratio scale,

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

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.

1.2 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*.

1.2.1 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.

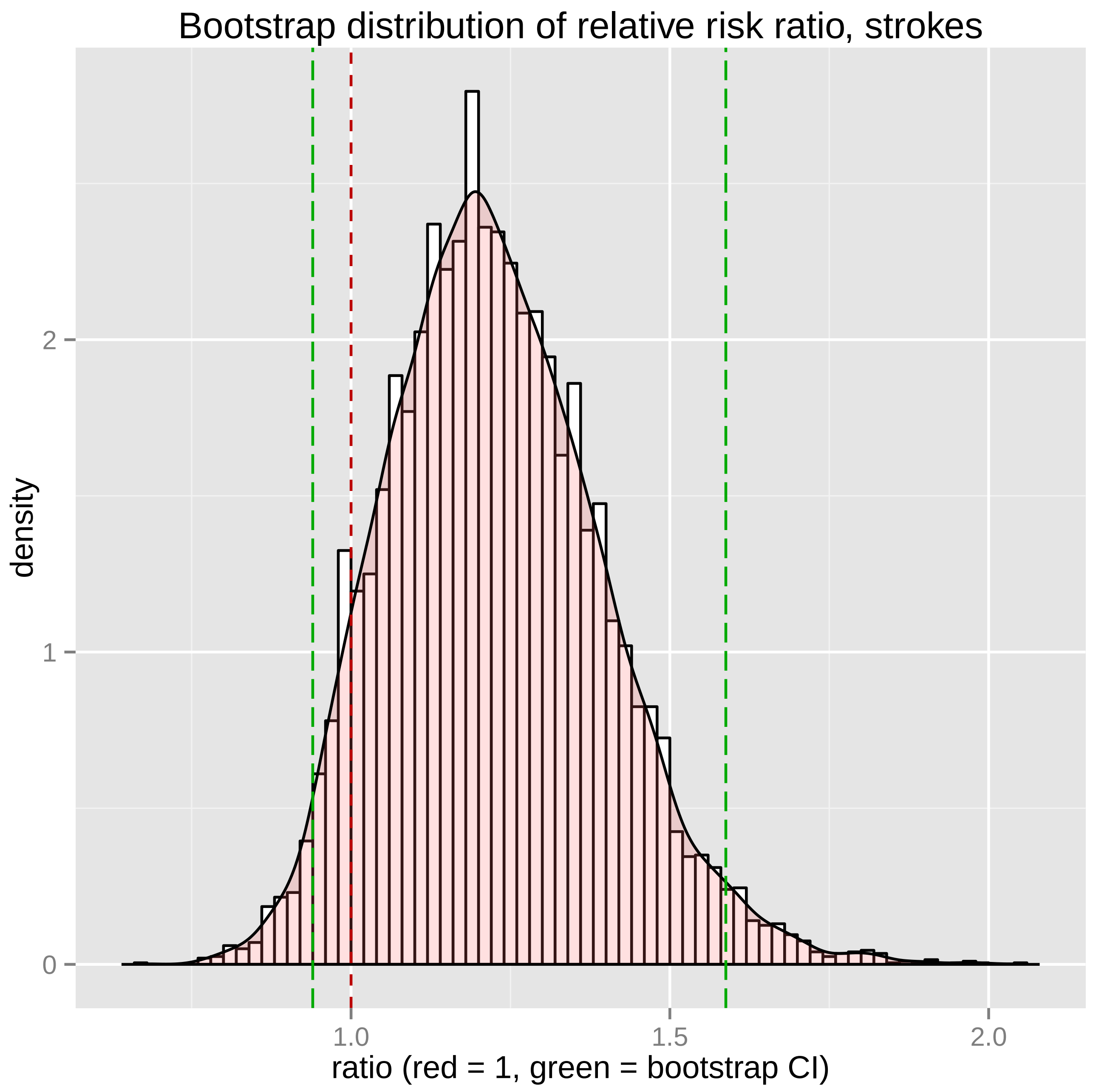
We draw with replacement a sample of 11037 items from the first population, and a sample of 11034 items from the second population. Each is called a *bootstrap sample*. From these we derive the bootstrap replicate of $\hat{\theta}$:

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

Repeat this process a large number of times, say 10000 times, and obtain 10000 *bootstrap replicates* $\hat{\theta}^*$. The summaries are in the code, followed by a histogram of bootstrap replicates, $\hat{\theta}^*$.

```
# sample size (n) and successes (s) for sample 1 (aspirin) and 2 (placebo)
n <- c(11037, 11034)
s <- c(  119,    98)
# data for samples 1 and 2, where 1 = success (stroke), 0 = failure (no stroke)
dat1 <- c(rep(1, s[1]), rep(0, n[1] - s[1]))
dat2 <- c(rep(1, s[2]), rep(0, n[2] - s[2]))
# draw R bootstrap replicates
R <- 10000
# init location for bootstrap samples
bs1 <- rep(NA, R)
bs2 <- rep(NA, R)
# draw R bootstrap resamples of proportions
for (i in 1:R) {
  # proportion of successes in bootstrap samples 1 and 2
  # (as individual steps for group 1:)
  resam1 <- sample(dat1, n[1], replace = TRUE)
  success1 <- sum(resam1)
  bs1[i] <- success1 / n[1]
  # (as one line for group 2:)
  bs2[i] <- sum(sample(dat2, n[2], replace = TRUE)) / n[2]
}
# bootstrap replicates of ratio estimates
rat <- bs1 / bs2
# sort the ratio estimates to obtain bootstrap CI
rat.sorted <- sort(rat)
# 0.025th and 0.975th quantile gives equal-tail bootstrap CI
CI.bs <- c(rat.sorted[round(0.025*R)], rat.sorted[round(0.975*R+1)])
CI.bs

## [1] 0.9399 1.5878
```

```
## Plot the bootstrap distribution with CI
# First put data in data.frame for ggplot()
dat.rat <- data.frame(rat)

library(ggplot2)
p <- ggplot(dat.rat, aes(x = rat))
p <- p + geom_histogram(aes(y=..density..)
                          , binwidth=0.02
                          , colour="black", fill="white")
  # Overlay with transparent density plot
p <- p + geom_density(alpha=0.2, fill="#FF6666")
# vertical line at 1 and CI
p <- p + geom_vline(aes(xintercept=1), colour="#BB0000", linetype="dashed")
p <- p + geom_vline(aes(xintercept=CI.bs[1]), colour="#00AA00", linetype="longdash")
p <- p + geom_vline(aes(xintercept=CI.bs[2]), colour="#00AA00", linetype="longdash")
p <- p + labs(title = "Bootstrap distribution of relative risk ratio, strokes")
p <- p + xlab("ratio (red = 1, green = bootstrap CI)")
print(p)

## Warning:  position_stack requires constant width:  output may be incorrect
```



In this simple case, the confidence interval derived from the bootstrap $(0.94, 1.588)$ 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.

1.2.2 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 such as 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. Implemention is done by Monte Carlo sampling.

The bootstrap in 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, two-sample t-test, mean 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	SE
Control:	52, 104, 146, 10, 51, 30, 40, 27, 46	9	56.22	14.14
Treatment:	94, 197, 16, 38, 99, 141, 23	7	86.86	25.24
		Difference:	30.63	28.93

Numerical and graphical summaries of the data are below. There seems to be a slight difference in variability between the two treatment groups.

```
treatment <- c(94, 197, 16, 38, 99, 141, 23)
control   <- c(52, 104, 146, 10, 51, 30, 40, 27, 46)
survive <- c(treatment, control)
group   <- c(rep("Treatment", length(treatment)), rep("Control", length(control)))
mice <- data.frame(survive, group)

library(plyr)
# ddply "dd" means the input and output are both data.frames
mice.summary <- ddply(mice,
                      "group",
                      function(X) {
                        data.frame( m = mean(X$survive),
                                    s = sd(X$survive),
                                    n = length(X$survive)
                                  )
                      }
                    )
# standard errors
mice.summary$se   <- mice.summary$s/sqrt(mice.summary$n)
# individual confidence limits
mice.summary$ci.l <- mice.summary$m - qt(1-.05/2, df=mice.summary$n-1) * mice.summary$se
mice.summary$ci.u <- mice.summary$m + qt(1-.05/2, df=mice.summary$n-1) * mice.summary$se
```

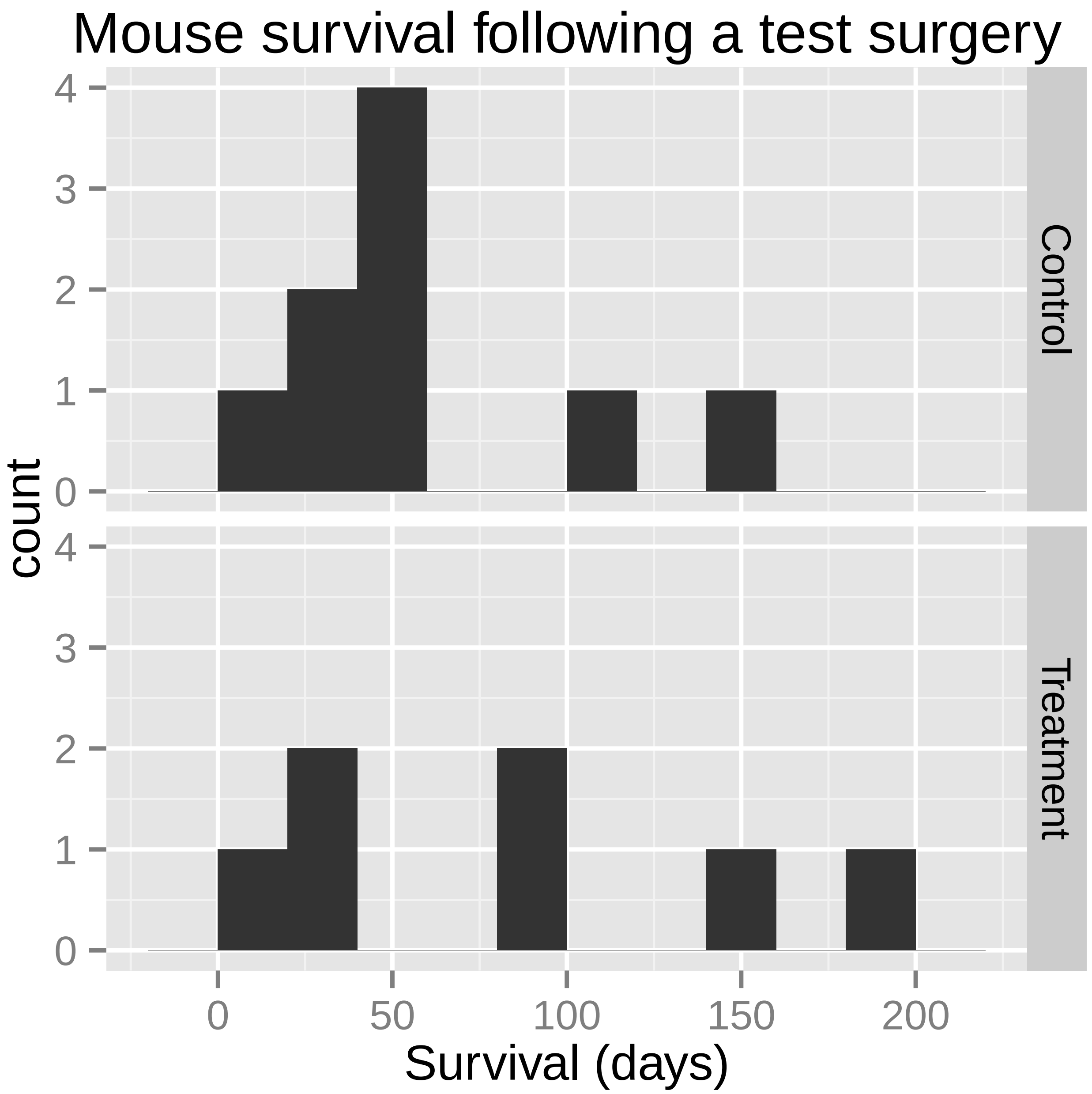
```
mice.summary
```

```
##       group     m     s n    se  ci.l   ci.u
## 1   Control 56.22 42.48 9 14.16 23.57  88.87
## 2 Treatment 86.86 66.77 7 25.24 25.11 148.61
```

```
diff(mice.summary$m)                                            #$
```

```
## [1] 30.63
```

```
# histogram using ggplot
p <- ggplot(mice, aes(x = survive))
p <- p + geom_histogram(binwidth = 20)
p <- p + facet_grid(group ~ .)
p <- p + labs(title = "Mouse survival following a test surgery") + xlab("Survival (days)")
print(p)
```



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.

```
t.test(survive ~ group, data = mice)

##
##  Welch Two Sample t-test
##
## data:  survive by group
## t = -1.059, df = 9.654, p-value = 0.3155
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -95.42  34.15
## sample estimates:
##   mean in group Control mean in group Treatment
##                   56.22                   86.86
```

But these are small samples, and the control sample 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 $\hat{\mu}$:

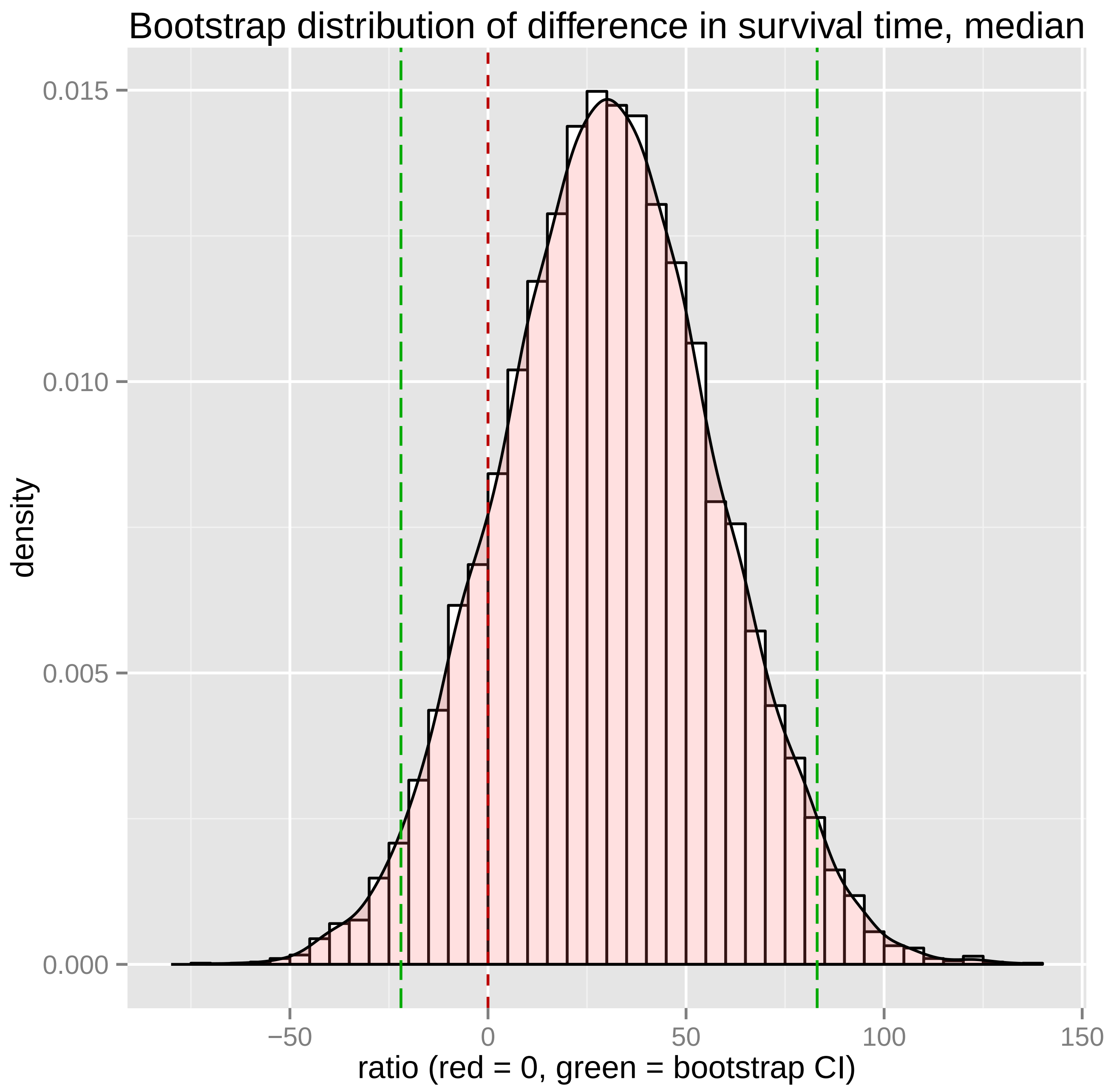
$$\hat{\mu}^* = \bar{x}^* - \bar{y}^*.$$

Repeat this process a large number of times, say 10000 times, and obtain 10000 *bootstrap replicates* $\hat{\mu}^*$. The summaries are in the code, followed by a histogram of bootstrap replicates, $\hat{\mu}^*$.

```
# draw R bootstrap replicates
R <- 10000
# init location for bootstrap samples
bs1 <- rep(NA, R)
bs2 <- rep(NA, R)
# draw R bootstrap resamples of means
```

```
for (i in 1:R) {
  bs2[i] <- mean(sample(control,   replace = TRUE))
  bs1[i] <- mean(sample(treatment, replace = TRUE))
}
# bootstrap replicates of difference estimates
bs.diff <- bs1 - bs2
sd(bs.diff)

## [1] 27

# sort the difference estimates to obtain bootstrap CI
diff.sorted <- sort(bs.diff)
# 0.025th and 0.975th quantile gives equal-tail bootstrap CI
CI.bs <- c(diff.sorted[round(0.025*R)], diff.sorted[round(0.975*R+1)])
CI.bs

## [1] -21.97  83.10
```

```
## Plot the bootstrap distribution with CI
# First put data in data.frame for ggplot()
dat.diff <- data.frame(bs.diff)

library(ggplot2)
p <- ggplot(dat.diff, aes(x = bs.diff))
p <- p + geom_histogram(aes(y=..density..)
                          , binwidth=5
                          , colour="black", fill="white")
  # Overlay with transparent density plot
p <- p + geom_density(alpha=0.2, fill="#FF6666")
# vertical line at 0 and CI
p <- p + geom_vline(aes(xintercept=0), colour="#BB0000", linetype="dashed")
p <- p + geom_vline(aes(xintercept=CI.bs[1]), colour="#00AA00", linetype="longdash")
p <- p + geom_vline(aes(xintercept=CI.bs[2]), colour="#00AA00", linetype="longdash")
p <- p + labs(title = "Bootstrap distribution of difference in survival time, median")
p <- p + xlab("ratio (red = 0, green = bootstrap CI)")
print(p)
```

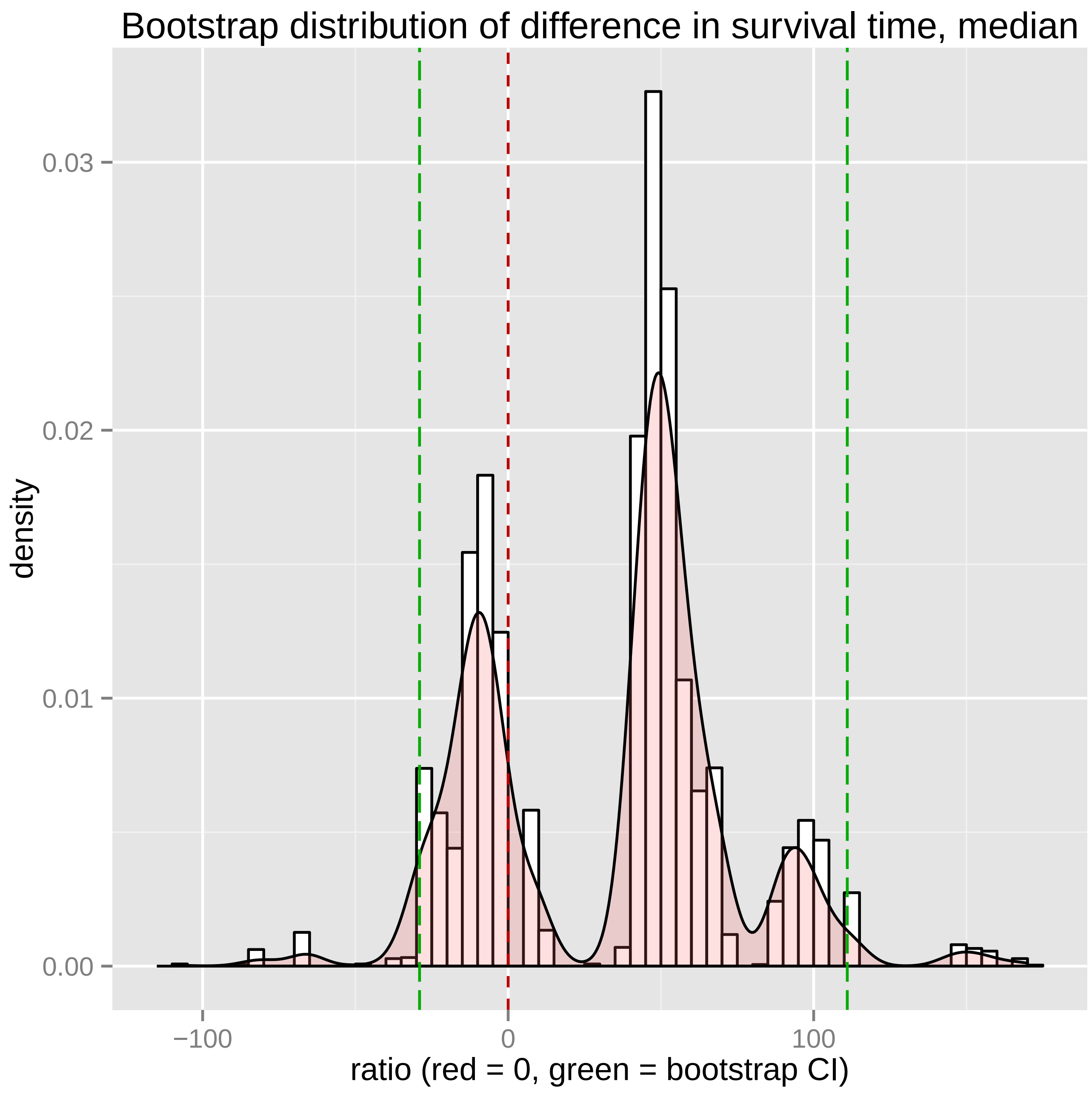


Example: Mouse survival, two-sample bootstrap, median For most statistics (such as the median) we don't have a formula for the limiting value of the standard error, but in fact no formula is needed. Instead, we use the numerical output of the bootstrap program. The summaries are in the code, followed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

$\hat{\eta}^*$.

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

```
sort(control)

## [1]  10  27  30  40  46  51  52 104 146

sort(treatment)

## [1]  16  23  38  94  99 141 197

# draw R bootstrap replicates
R <- 10000
# init location for bootstrap samples
bs1 <- rep(NA, R)
bs2 <- rep(NA, R)
# draw R bootstrap resamples of medians
for (i in 1:R) {
  bs2[i] <- median(sample(control,   replace = TRUE))
  bs1[i] <- median(sample(treatment, replace = TRUE))
}
# bootstrap replicates of difference estimates
bs.diff <- bs1 - bs2
sd(bs.diff)

## [1] 40.43

# sort the difference estimates to obtain bootstrap CI
diff.sorted <- sort(bs.diff)
# 0.025th and 0.975th quantile gives equal-tail bootstrap CI
CI.bs <- c(diff.sorted[round(0.025*R)], diff.sorted[round(0.975*R+1)])
CI.bs

## [1] -29 111
## Plot the bootstrap distribution with CI
# First put data in data.frame for ggplot()
dat.diff <- data.frame(bs.diff)

library(ggplot2)
p <- ggplot(dat.diff, aes(x = bs.diff))
p <- p + geom_histogram(aes(y=..density..)
                          , binwidth=5
                          , colour="black", fill="white")
  # Overlay with transparent density plot
p <- p + geom_density(alpha=0.2, fill="#FF6666")
# vertical line at 0 and CI
p <- p + geom_vline(aes(xintercept=0), colour="#BB0000", linetype="dashed")
p <- p + geom_vline(aes(xintercept=CI.bs[1]), colour="#00AA00", linetype="longdash")
p <- p + geom_vline(aes(xintercept=CI.bs[2]), colour="#00AA00", linetype="longdash")
p <- p + labs(title = "Bootstrap distribution of difference in survival time, median")
p <- p + xlab("ratio (red = 0, green = bootstrap CI)")
print(p)
```

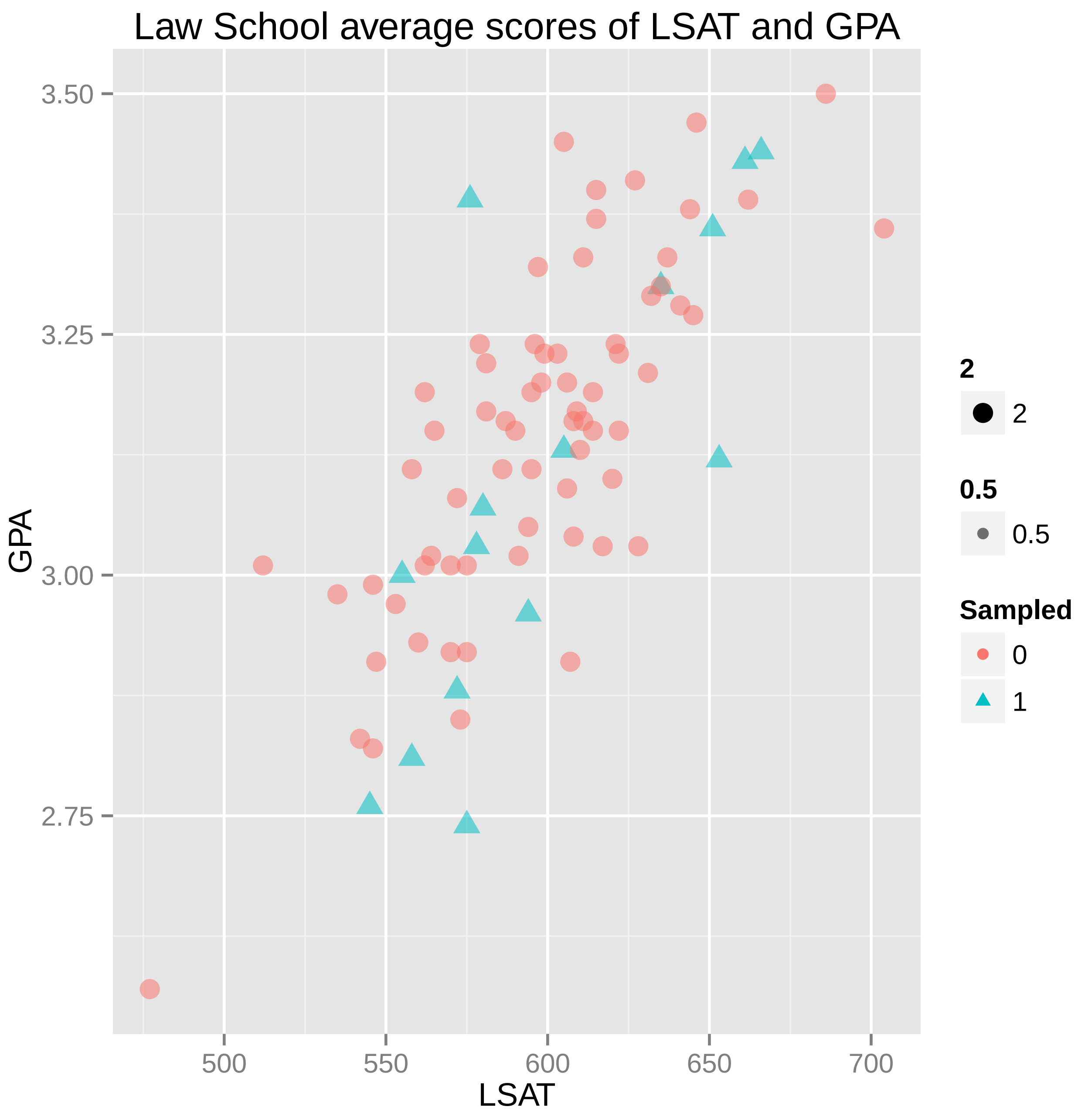


1.2.3 Comparing bootstrap sampling distribution from population and sample

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			

```
School  <- 1:82

LSAT    <- c(622, 542, 579, 653, 606, 576, 620, 615, 553, 607, 558, 596, 635,
             581, 661, 547, 599, 646, 622, 611, 546, 614, 628, 575, 662, 627,
             608, 632, 587, 581, 605, 704, 477, 591, 578, 572, 615, 606, 603,
             535, 595, 575, 573, 644, 545, 645, 651, 562, 609, 555, 586, 580,
             594, 594, 560, 641, 512, 631, 597, 621, 617, 637, 572, 610, 562,
             635, 614, 546, 598, 666, 570, 570, 605, 565, 686, 608, 595, 590,
             558, 611, 564, 575)

GPA     <- c(3.23, 2.83, 3.24, 3.12, 3.09, 3.39, 3.10, 3.40, 2.97, 2.91, 3.11,
             3.24, 3.30, 3.22, 3.43, 2.91, 3.23, 3.47, 3.15, 3.33, 2.99, 3.19,
             3.03, 3.01, 3.39, 3.41, 3.04, 3.29, 3.16, 3.17, 3.13, 3.36, 2.57,
             3.02, 3.03, 2.88, 3.37, 3.20, 3.23, 2.98, 3.11, 2.92, 2.85, 3.38,
             2.76, 3.27, 3.36, 3.19, 3.17, 3.00, 3.11, 3.07, 2.96, 3.05, 2.93,
             3.28, 3.01, 3.21, 3.32, 3.24, 3.03, 3.33, 3.08, 3.13, 3.01, 3.30,
             3.15, 2.82, 3.20, 3.44, 3.01, 2.92, 3.45, 3.15, 3.50, 3.16, 3.19,
             3.15, 2.81, 3.16, 3.02, 2.74)

Sampled <- c(0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,
             1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1)

# law = population
law <- data.frame(School, LSAT,  GPA, Sampled)
law$Sampled <- factor(law$Sampled)
# law.sam = sample
law.sam <- subset(law, Sampled == 1)

library(ggplot2)
p <- ggplot(law, aes(x = LSAT, y = GPA))
p <- p + geom_point(aes(colour = Sampled, shape = Sampled, alpha = 0.5, size = 2))
p <- p + labs(title = "Law School average scores of LSAT and GPA")
print(p)
```



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.

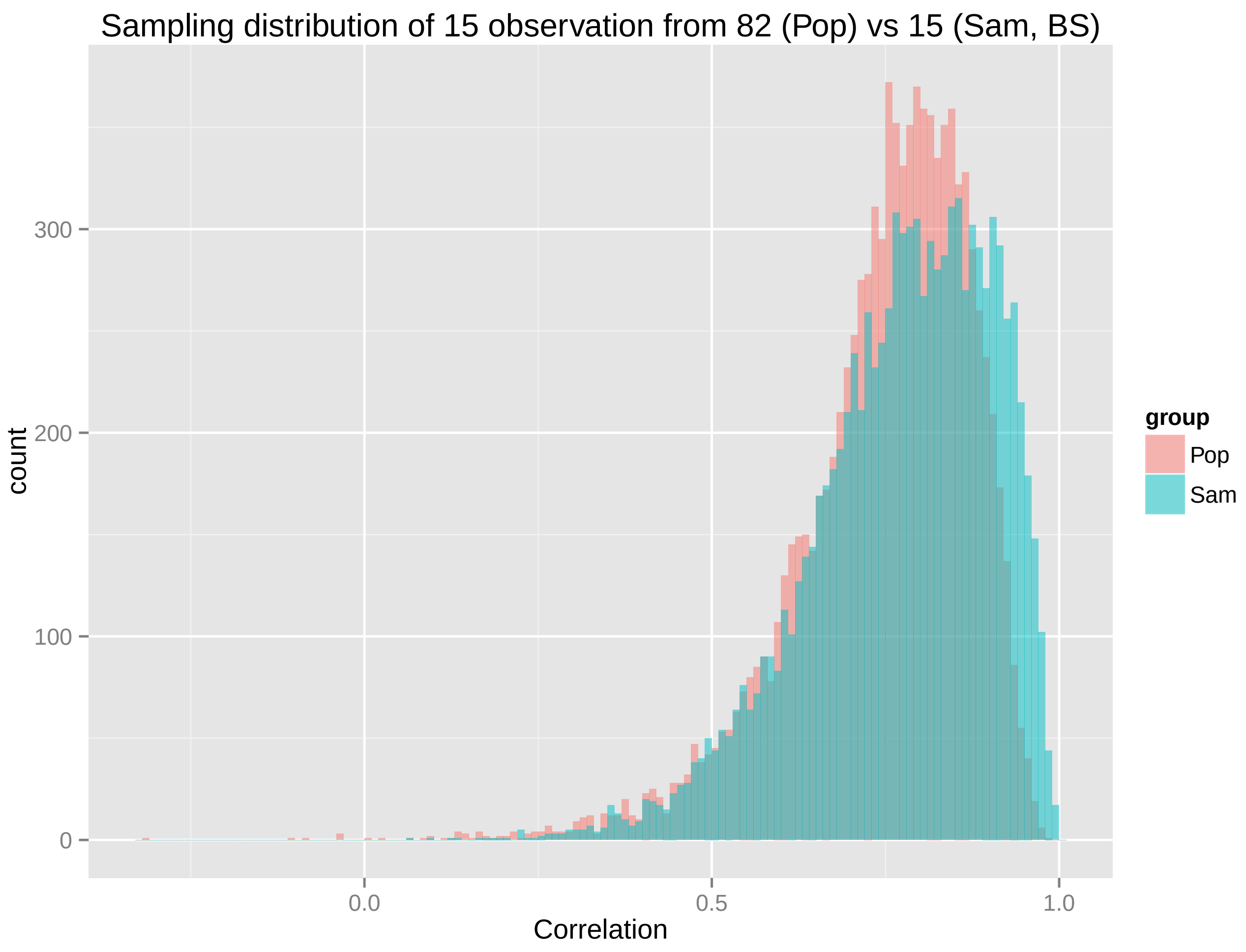
The code below does that, followed by two histograms. In this case, the histograms are noticeably non-normal, having a long tail toward the left. Inferences based on the normal curve are suspect when the bootstrap histogram is markedly non-normal. The histogram on the left is the nonparametric bootstrap sampling distribution using only the $n = 15$ sampled schools with 10000 bootstrap replicates of $\widehat{\text{corr}}(x^*)$. The histogram on the right is the true sampling distribution using 10000 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.

```r
# draw R bootstrap replicates
R <- 10000
# init location for bootstrap samples
bs.pop <- rep(NA, R)
bs.sam <- rep(NA, R)
# draw R bootstrap resamples of medians
for (i in 1:R) {
  # sample() draws indicies then bootstrap correlation of LSAT and GPA
  # population
  bs.pop[i] = cor(law    [sample(seq(1,nrow(law    )), nrow(law.sam)
                                 , replace = TRUE), 2:3])[1, 2]
  # sample
  bs.sam[i] = cor(law.sam[sample(seq(1,nrow(law.sam)), nrow(law.sam)
                                 , replace = TRUE), 2:3])[1, 2]
}

# sort the difference estimates to obtain bootstrap CI
diff.sorted <- sort(bs.pop)
```

```
# 0.025th and 0.975th quantile gives equal-tail bootstrap CI
CI.bs.pop <- c(diff.sorted[round(0.025*R)], diff.sorted[round(0.975*R+1)])
# population correlation
cor(law    [, c(2,3)])[1,2]

## [1] 0.76

CI.bs.pop

## [1] 0.4297 0.9271

sd(bs.pop)

## [1] 0.1295


# sort the difference estimates to obtain bootstrap CI
diff.sorted <- sort(bs.sam)
# 0.025th and 0.975th quantile gives equal-tail bootstrap CI
CI.bs.sam <- c(diff.sorted[round(0.025*R)], diff.sorted[round(0.975*R+1)])
# sample correlation
cor(law.sam[, c(2,3)])[1,2]

## [1] 0.7764

CI.bs.sam

## [1] 0.4638 0.9638

sd(bs.sam)

## [1] 0.1335
```

```
law.bs.df <- data.frame(corr = c(bs.pop, bs.sam), group = c(rep("Pop",R),rep("Sam",R)))
# histogram using ggplot
library(ggplot2)
p <- ggplot(law.bs.df, aes(x = corr, fill=group))
p <- p + geom_histogram(binwidth = .01, alpha = 0.5, position="identity")
p <- p + labs(title = "Sampling distribution of 15 observation from 82 (Pop) vs 15 (Sam, BS)")
```

```
          xlab("Correlation")
print(p)
```



1.3 Background and notation

Let[2] $X_1, \ldots, X_n$ be iid (independent and identically distributed) random variables with density (or mass function) $f(t)$ and cumulative distribution function (cdf) $F(t)$. For simplicity assume the Xs are scaler random variables.

Suppose we are interested in some feature (parameter/expectation) of the distribution, say θ. Givens and Hoeting use "functional" notation to identify this feature:

$$\theta = T(F)$$

where θ is a function T of the distribution indexed by cdf $F(t)$. In conjunction with this notation, it is convenient to use **Lebesgue-Stieltjes integrals** to represent functionals. For example, if

$$\begin{aligned} \theta = \mathrm{E}[X_i] &= \int t f(t)\,\mathrm{d}t && \text{if } f(t) \text{ density} \\ &= \sum_{i=1}^{s} t_i f(t_i) && \text{if } f(t) \text{ discrete with probability } f(t_i) \text{ at } t_i \\ &= \int t\,\mathrm{d}F(t). \end{aligned}$$

That is, the Lebesgue-Stieltjes integral

$$\theta = \int t\,\mathrm{d}F(t)$$

corresponds to the expressions above it for continuous and discrete random variables.

[2]References for this section include Givens and Hoeting (Chapter 9) or Davison and Hinkley (Chapter 2).

As another example, if

$$\begin{aligned}\theta \;=\; \Pr[X_i \geq c] \;&=\; \int_c^\infty f(t)\,\mathrm{d}t \;=\; \int 1_{(t\geq c)} f(t)\,\mathrm{d}t \qquad f(t)\text{ continuous}\\ &=\; \sum_{t_i : t_i \geq c} f(t_i) \qquad f(t)\text{ discrete}\end{aligned}$$

then

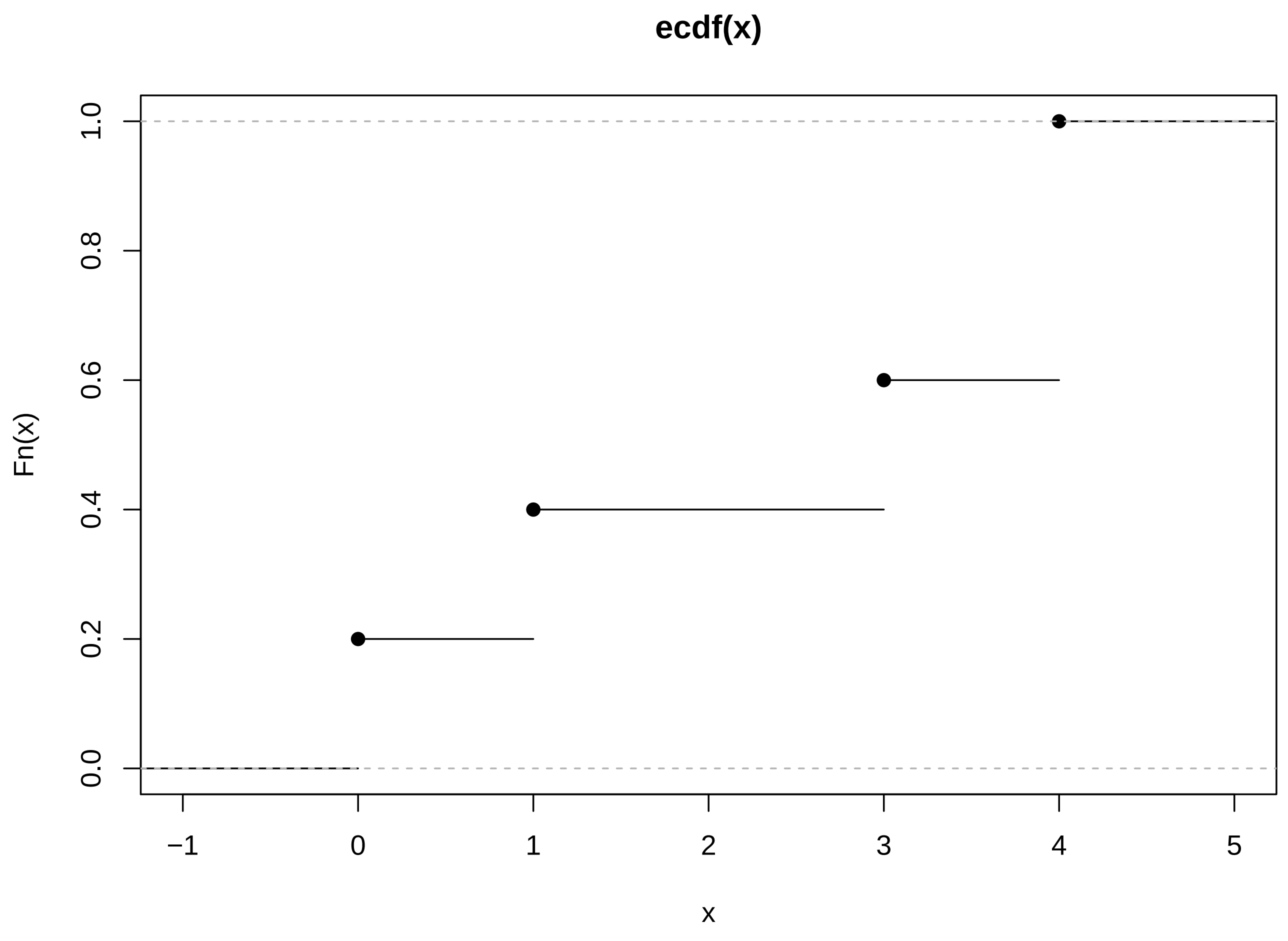
$$\theta \;=\; \int_c^\infty \mathrm{d}F(t) \;=\; \int 1_{(t\geq c)}\,\mathrm{d}F(t).$$

If you feel uncomfortable with this formality, just think of $\mathrm{d}F(t) = f(t)\,\mathrm{d}t$ in integral representation for continuous distributions.

This notation is actually convenient for deriving distribution theory for estimators, and in particular in the context of bootstrapping. Let $\underset{\sim}{x} = \{x_1, x_2, \ldots, x_n\}$ denote the entire sample, and let

$$\begin{aligned}\hat{F}(t) \;&=\; \text{empirical distribution function}\\ &=\; \frac{1}{n}\sum_{j=1}^{n} 1_{(x_j \leq t)} \;=\; \frac{\#\; x_j\text{s} \leq t}{n}.\end{aligned}$$

```r
plot.ecdf(c(4, 1, 3, 0, 4))
```



The empirical cdf $\hat{F}(t)$ is a non-parametric estimator of $F(t)$. In particular, if we think of t as fixed, then

$$\begin{aligned} n\hat{F}(t) &= (\#\ x_j\text{s} \leq t) \\ &\sim \text{Binomial}(n, p) \end{aligned}$$

where

$$p = \Pr(x_j \leq t) = F(t).$$

Thus, for example,

$$\begin{aligned}
\mathrm{E}[\hat{F}(t)] &= \frac{1}{n}\mathrm{E}[n\hat{F}(t)] = \frac{1}{n}nF(t) = F(t) \qquad \text{and} \\
\mathrm{Var}[\hat{F}(t)] &= \frac{1}{n^2}\mathrm{Var}[n\hat{F}(t)] = \frac{1}{n^2}nF(t)\{1-F(t)\} = \frac{F(t)\{1-F(t)\}}{n},
\end{aligned}$$

and further, by the delta method, for fixed t

$$\begin{aligned}
\sqrt{n}\{\hat{F}(t)-F(t)\} &\dot{\sim} \mathrm{Normal}(0, F(t)\{1-F(t)\}) \qquad \text{or} \\
\hat{F}(t) &\dot{\sim} \mathrm{Normal}(F(t), \frac{F(t)\{1-F(t)\}}{n}).
\end{aligned}$$

It is important to realize that $\hat{F}(t)$ is a bona-fide distribution function, corresponding to a random variable X^* that assumes values $x_1, x_2, \ldots, x_n$ (the observed values of $X_1, X_2, \ldots, X_n$) each with probability $1/n$.

Further, the feature or functional of interest, θ, is naturally estimated via

$$\begin{aligned}
\hat{\theta} = T(\hat{F}) &= \frac{1}{n}\sum_{j=1}^{n} x_j = \bar{x} \\
&\qquad \text{if } \theta = \mathrm{E}(x_i) = \int t\,\mathrm{d}F(t) \\
&= \int 1_{(t\geq c)}\,\mathrm{d}\hat{F}(t) = \frac{1}{n}\sum_{j=1}^{n} 1_{(x_j \leq c)} = \frac{\#\, x_j\mathrm{s} \leq c}{n} \\
&\qquad \text{if } \theta = \Pr(x_i \geq c) = \int 1_{(t\geq c)}\,\mathrm{d}F(t).
\end{aligned}$$

A more complicated example of a functional might be

$$\begin{aligned}
S(F) &= \int (t-\theta)^2\,\mathrm{d}F(t) \qquad \text{where } \theta = \int t\,\mathrm{d}F(t) \\
&= \mathrm{E}(x_i-\theta)^2 \\
&= \mathrm{Var}(x_i)
\end{aligned}$$

which may be estimated by

$$\begin{aligned} S(\hat{F}) &= \int (t-\hat{\theta})^2 \,\mathrm{d}\hat{F}(t) \qquad \text{where } \hat{\theta} = \int t\,\mathrm{d}\hat{F}(t) = \bar{x} \\ &= \frac{1}{n}\sum_{j=1}^{n}(x_j-\bar{x})^2 \\ &= \text{“divide-by-n” version of sample variance.} \end{aligned}$$

Questions of statistical inference are usually posed in terms of the estimator

$$\hat{\theta} = T(\hat{F})$$

or some

$$R(\underset{\sim}{x}, F) = \text{function of sample } \underset{\sim}{x} \text{ and } F.$$

For example, $R(\underset{\sim}{x}, F)$ might correspond to the "t-statistic"

$$R(\underset{\sim}{x}, F) = \frac{T(\hat{F}) - T(F)}{\sqrt{S(\hat{F})}}$$

where $\hat{F}$ depends on $\underset{\sim}{x}$ and where

$$\begin{aligned} T(F) &= \int t\,\mathrm{d}F(t) = \mathrm{E}(x_i), \\ T(\hat{F}) &= \int t\,\mathrm{d}\hat{F}(t) = \bar{x} \\ S(\hat{F}) &= \int (t-\hat{\theta})^2\,\mathrm{d}\hat{F}(t) = \frac{1}{n}\sum_{j=1}^{n}(x_j-\bar{x})^2. \end{aligned}$$

A primary question might be "what is the distribution of $R(\underset{\sim}{x}, F)$?" This may be intractable, unknown, or depend on F which is unknown.

The empirical or nonparametric (NP) bootstrap works as follows. If we have a collection of **iid** random variables $\underset{\sim}{x} = \{x_1, x_2, \ldots, x_n\}$ with cdf $F(t)$, then the probability distribution of $R(\underset{\sim}{x}, F)$ can be approximated by the probability distribution of $R(\underset{\sim}{x}^*, \hat{F})$ where $\underset{\sim}{x}^* = \{x_1^*, x_2^*, \ldots, x_n^*\}$ are iid random variables with cdf $\hat{F}(t)$.

Remarks

- Idea is simple, yet powerful — and implications can be very subtle.
- If n is small, distribution of $R(\underset{\sim}{x}^*, \hat{F})$ can be computed exactly in certain cases, and otherwise approximated using **resampling** (that is, repeated bootstrap samples).
- A bootstrap sample $\underset{\sim}{x}^* = \{x_1^*, x_2^*, \ldots, x_n^*\}$ from $\hat{F}(t)$ is equivalent to **sampling with replacement** n data values from the original sample $\underset{\sim}{x} = \{x_1, x_2, \ldots, x_n\}$. Thus, the bootstrap approximates the unknown $F(t)$ by the empirical cdf $\hat{F}(t)$, and then uses repeated samples from the original sample to estimate the distribution of a statistic (that is, treats the original sample as the population).

Example Suppose X_1, X_2, X_3 are iid with cdf $F(t)$ and define

$$\theta = \mathrm{E}[X_i] = \int t \, \mathrm{d}F(t) \equiv T(F).$$

Two distribution we are interested in (and they are related) are the distribution of

$$\hat{\theta} = \int t \, \mathrm{d}\hat{F}(t) = \bar{x} = T(\hat{F}) \tag{1.1}$$

$$\hat{\theta} - \theta = \int t \, \mathrm{d}\hat{F}(t) - \int t \, \mathrm{d}F(t) = T(\hat{F}) - T(F) = \bar{x} - \theta. \tag{1.2}$$

Note that if we use notation $R(\underset{\sim}{x}, F)$, then (1.1)

$$\hat{\theta} \;\equiv\; R(\underset{\sim}{x}, F)$$

is a function of $\underset{\sim}{x}$ through $\hat{F}(t)$ (depending on $\underset{\sim}{x}$), but is not a function of F. While, if (1.2)

$$\hat{\theta} - \theta \;=\; T(\hat{F}) - T(F) \;=\; R(\underset{\sim}{x}, F) \tag{1.3}$$

then this quantity depends on $\underset{\sim}{x}$ and F.

Let us consider (1.1) first. Suppose our observed sample is $x_1 = 6, x_2 = 1, x_3 = 2$. The empirical cdf $\hat{F}(t)$ places mass $1/3$ at each of the three points: 1, 2, and 6.

A bootstrap sample $\underset{\sim}{x}^* = \{x_1^*, x_2^*, x_3^*\}$ is a sample with replacement of size three from $\{1, 2, 6\}$, or equivalently the x_i^* are independent with

$$x_i^* \;=\; 1, 2, \text{ or } 6 \text{ with probability } 1/3 \text{ each}$$

There are $3^3 = 27$ possible bootstrap samples, each with probability $1/27$. However, only the ordered samples are needed to generate the bootstrap distribution for

$$\begin{aligned} \hat{\theta}^* \;&=\; T(\hat{F}^*) \;=\; R(\underset{\sim}{x}^*, \hat{F}) \\ &=\; \bar{x}^* \;\equiv\; \text{mean of bootstrap sample} \end{aligned}$$

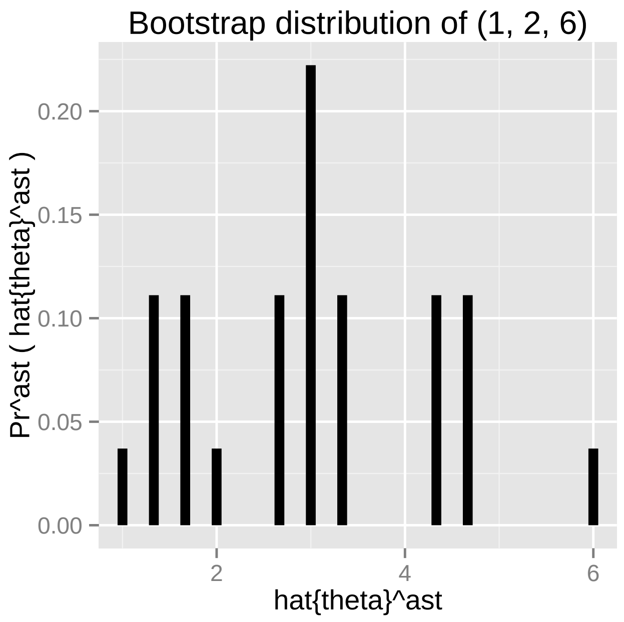
where $\hat{\theta}^*$ is the estimate computed from the bootstrap sample, $\hat{F}^*$ is the empirical cdf of bootstrap sample $\underset{\sim}{x}^* = \{x_1^*, x_2^*, x_3^*\}$, and each $\underset{\sim}{x}_i^*$ has distribution $\hat{F}$. The bootstrap distribution is given in the table below.

```r
# original sample
x <- c(1,2,6)
# Cartesian product of the three observations has 3^3 combinations
xast <- expand.grid(x1ast = x, x2ast = x, x3ast = x)
```

```
# order each row independently
xast.sort <- t(apply(xast, 1, sort))
rownames(xast.sort) <- 1:nrow(xast.sort)
# combine into a single column, with mean
xast.1col <- data.frame(xast.combine =
                          paste(xast.sort[,1], xast.sort[,2], xast.sort[,3], sep=" ")
                        )
# select the unique ones
xast.unique <- unique(xast.1col)
xast.sample <- xast[as.numeric(rownames(xast.unique)),]
# calculate the mean of the unique ones
thetahatast <- rowMeans(xast.sort[as.numeric(rownames(xast.unique)),])
# count up how many of each there are, and divide by n for a probability
Prast.thetahatast <- as.vector(xtabs( ~ xast.combine, xast.1col))/nrow(xast)
# put together as a data.frame
xast.summary <- data.frame(xast.sample, xast.unique, thetahatast, Prast.thetahatast)
# display the data.frame
xast.summary

##    x1ast x2ast x3ast xast.combine thetahatast Prast.thetahatast
## 1      1     1     1        1 1 1       1.000           0.03704
## 2      2     1     1        1 1 2       1.333           0.11111
## 3      6     1     1        1 1 6       2.667           0.11111
## 5      2     2     1        1 2 2       1.667           0.11111
## 6      6     2     1        1 2 6       3.000           0.22222
## 9      6     6     1        1 6 6       4.333           0.11111
## 14     2     2     2        2 2 2       2.000           0.03704
## 15     6     2     2        2 2 6       3.333           0.11111
## 18     6     6     2        2 6 6       4.667           0.11111
## 27     6     6     6        6 6 6       6.000           0.03704
```

	x1ast	x2ast	x3ast	xast.combine	thetahatast	Prast.thetahatast
1	1.0000	1.0000	1.0000	1 1 1	1.0000	0.0370
2	2.0000	1.0000	1.0000	1 1 2	1.3333	0.1111
3	6.0000	1.0000	1.0000	1 1 6	2.6667	0.1111
5	2.0000	2.0000	1.0000	1 2 2	1.6667	0.1111
6	6.0000	2.0000	1.0000	1 2 6	3.0000	0.2222
9	6.0000	6.0000	1.0000	1 6 6	4.3333	0.1111
14	2.0000	2.0000	2.0000	2 2 2	2.0000	0.0370
15	6.0000	2.0000	2.0000	2 2 6	3.3333	0.1111
18	6.0000	6.0000	2.0000	2 6 6	4.6667	0.1111
27	6.0000	6.0000	6.0000	6 6 6	6.0000	0.0370

```r
library(ggplot2)
p <- ggplot(xast.summary, aes(x = thetahatast, y = Prast.thetahatast))
p <- p + geom_segment(aes(yend=0, xend=thetahatast), size=2)
p <- p + labs(title = "Bootstrap distribution of (1, 2, 6)")
p <- p + ylab("Pr^ast ( hat{theta}^ast )")
p <- p + xlab("hat{theta}^ast")
print(p)
```



The bootstrap distribution of $\hat{\theta}^*$ approximates the distribution of $\hat{\theta}$.

Now, let us move on to (1.2), where we are interested in the distribution of

$$\hat{\theta} - \theta \;=\; T(\hat{F}) - T(F) \;\equiv\; R(\underset{\sim}{x}, \hat{F}).$$

Though θ is unknown, the distribution of $\hat{\theta} - \theta$ is approximated by the bootstrap distribution of

$$\hat{\theta}^* - \hat{\theta} \;=\; T(\hat{F}^*) - T(\hat{F}) \;\equiv\; R(\underset{\sim}{x}^*, \hat{F}^*).$$

The value of $\hat{\theta}$ is known[3]: $\hat{\theta} = 3$, so the bootstrap distribution of $\hat{\theta}^* - \hat{\theta}$ is just the distribution of $\hat{\theta}^*$ shifted leftwards by $\hat{\theta} = 3$.

```
xast.summary$thetahatastdiff <- xast.summary$thetahatast - mean(x)
```

	thetahatastdiff	Prast.thetahatast
1	-2.0000	0.0370
2	-1.6667	0.1111
3	-0.3333	0.1111
5	-1.3333	0.1111
6	0.0000	0.2222
9	1.3333	0.1111
14	-1.0000	0.0370
15	0.3333	0.1111
18	1.6667	0.1111
27	3.0000	0.0370

Suppose instead of a sample of three, we had a sample $\underset{\sim}{x} = \{x_1, x_2, \dots, x_n\}$ of arbitrary size, n. If the x_is are distinct, the number of bootstrap samples $\underset{\sim}{x}^* = \{x_1^*, x_2^*, \dots, x_n^*\}$ is large, n^n. In this case, the bootstrap distribution

[3]Actually, better to think of this as $\hat{\theta}$ is fixed relative to the bootstrap distribution, which samples from $\hat{F}(t)$, which is fixed. Hence, $\hat{\theta} = T(\hat{F})$ is fixed relative to the bootstrap distribution.

of any statistic $\hat{\theta}^*$ would be impossible to generate, but trivial to approximate via Monte Carlo, by simply generating repeated bootstrapped samples. In particular, if we generate B independent bootstrap samples:

$$\begin{aligned}
\underset{\sim}{x}_1^* &= \{x_{11}^*, x_{12}^*, \ldots, x_{1n}^*\} \text{ giving } \hat{\theta}_1^* \\
\underset{\sim}{x}_2^* &= \{x_{21}^*, x_{22}^*, \ldots, x_{2n}^*\} \text{ giving } \hat{\theta}_2^* \\
&\vdots \\
\underset{\sim}{x}_B^* &= \{x_{B1}^*, x_{B2}^*, \ldots, x_{Bn}^*\} \text{ giving } \hat{\theta}_B^*
\end{aligned}$$

Where each is a with replacement sample from $\underset{\sim}{x}^* = \{x_1^*, x_2^*, \ldots, x_n^*\}$, then the $\{\hat{\theta}_i^*\}$s are an iid sample from the bootstrap distribution of $\hat{\theta}^*$. Thus, the observed distribution of the $\{\hat{\theta}_i^*\}$s can be used to approximate or estimate any property of the bootstrapp distribution. As $B \to \infty$, our estimates of the bootstrap distribution converges to "true values".

Keep in mind that even if you know the bootstrap distribution of $\hat{\theta}^*$ it is still an **approximation only** to the distribution of $\hat{\theta}$! The same idea applies when bootstrapping $\hat{\theta}^* - \hat{\theta}$.

R has a variety of tools for bootstrapping, including functions in the `boot` library. Also, as we have already seen, the `sample()` function allows you to sample with or without replacement from a vector.

Example, nonparametric BS of CV Suppose $X_1, X_2, \ldots, X_n$ are iid from a distribution with cdf $F(t)$, and we are interested in estimating the population coefficient-of-variation

$$\text{CV} = 100\frac{\sigma}{\mu}$$

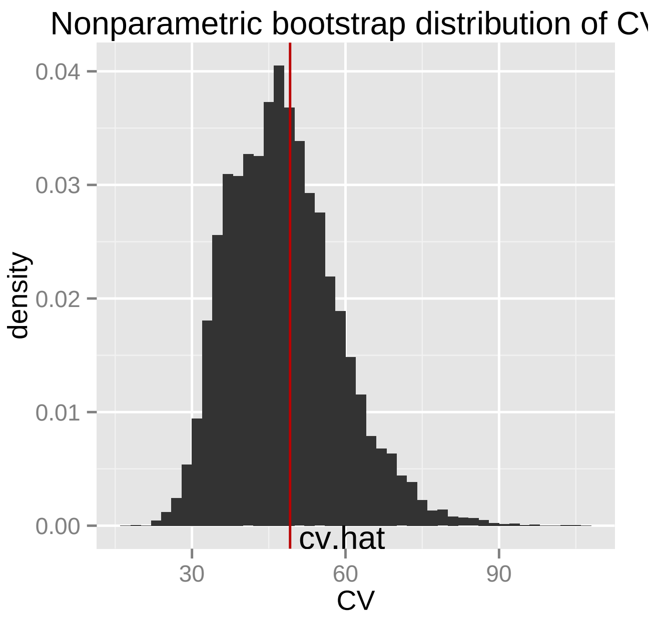
where $\sigma^2 = \text{Var}(X_i)$ and $\mu = \text{E}(X_i)$. That is, the CV (in %) tells you about how large the standard deviation in the population is relative to the size of the population mean.

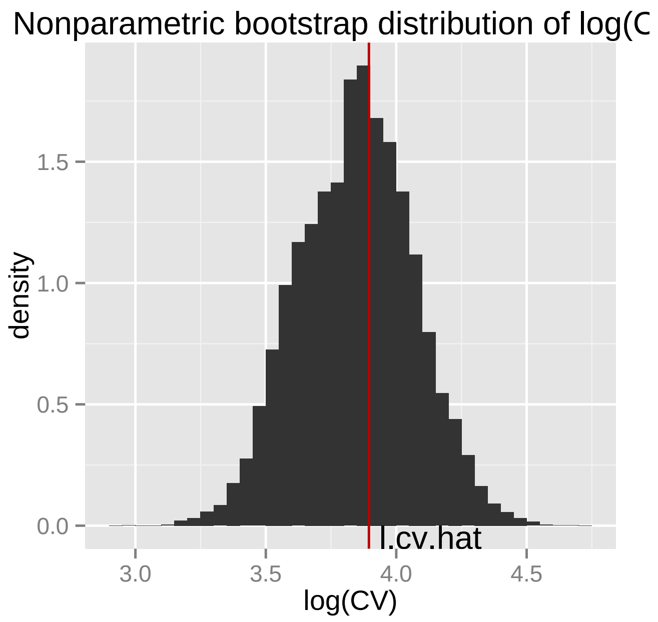
Let's assume the population distribution is Normal(4, 4), giving a population coefficient-of-variation CV $= 100 \times \sqrt{4}/4 = 0.50\%$. We assume this fact is unknown to the analyst, who wants to estimate the CV. Assume she draws a sample of size $n = 20$. Let's estimates the sampling distribution of the CV using a nonparametric (resample with replacement) bootstrap.

```r
# sample size
n <- 20;
# draw sample
x <- rnorm(n, mean = 4, sd = sqrt(4))
# correction factor to use "divide-by-n" variance
n1.n <- sqrt((n - 1) / n)

# Sample summaries
sd.mle   <- n1.n * sd(x)  # sd mle
mu.hat   <- mean(x)       # mean
cv.hat   <- 100 * sd.mle / mu.hat  # estimate of the CV
l.cv.hat <- log(cv.hat)          # log of the CV

# print values with column names
data.frame(sd.mle, mu.hat, cv.hat, l.cv.hat)


##   sd.mle mu.hat cv.hat l.cv.hat
## 1  2.116  4.303  49.17    3.895



# Nonparametric bootstrap
R <- 1e4
# initialize a vector of NAs to hold the CVs as they are calculated
cv.bs <- rep(NA, R)
for (i.R in 1:R) {
  # resample with replacement
  x.ast <- sample(x, replace = TRUE)
  # calculate the CV of each resample
  cv.bs[i.R] <- 100 * n1.n * sd(x.ast) / mean(x.ast)
}
l.cv.bs <- log(cv.bs) # log CV

# bs summaries in data.frame
bs.sum <- data.frame(cv.bs, l.cv.bs)
```

Note that there's a faster version of the above code which draws all the samples in one step and calculates row standard deviations and row means of a matrix of samples. Try it.

```
# CV
library(ggplot2)
p <- ggplot(bs.sum, aes(x = cv.bs))
p <- p + geom_histogram(aes(y = ..density..), binwidth=2)
p <- p + labs(title = "Nonparametric bootstrap distribution of CV")
p <- p + geom_vline(aes(xintercept=cv.hat), colour="#BB0000", linetype="solid")
p <- p + geom_text(data = data.frame(NA)
                   , aes(label = "cv.hat", x=cv.hat, y=0, hjust=-0.1, vjust=1))
p <- p + xlab("CV")
print(p)

# log(CV)
library(ggplot2)
p <- ggplot(bs.sum, aes(x = l.cv.bs))
p <- p + geom_histogram(aes(y = ..density..), binwidth=.05)
p <- p + labs(title = "Nonparametric bootstrap distribution of log(CV)")
p <- p + geom_vline(aes(xintercept=l.cv.hat), colour="#BB0000", linetype="solid")
p <- p + geom_text(data = data.frame(NA)
                   , aes(label = "l.cv.hat", x=l.cv.hat, y=0, hjust=-0.1, vjust=1))
p <- p + xlab("log(CV)")
print(p)
```





For the sample, the CV is about 55%, which is fairly close to the population CV of 50%. The bootstrap distribution of CV is skewed to the left while the bootstrap distribution of log(CV) is skewed to the right.

1.3.1 Parametric bootstrap

Suppose $X_1, X_2, \ldots, X_n$ are iid from a distribution with cdf $F_\tau(t)$ that depends on a parameter τ, which could be a scaler or vector. Assume we are interested in the distribution of $R(\underset{\sim}{x}, F_\tau)$, where as before $\underset{\sim}{x}^* = \{x_1^*, x_2^*, \ldots, x_n^*\}$. In the parametric bootstrap we assume the model holds, estimate τ based on the data, typically by maximum likelihood (ML), then estimate the distribution of $R(\underset{\sim}{x}, F_\tau)$ with the distribution of $R(\underset{\sim}{x}^*, F_{\hat{\tau}})$. Here $\hat{\tau}$ is the estimate of τ.

The only wrinkle with the parametric bootstrap is that the bootstrap samples are from the distribution $F_{\hat{\tau}}$, which is an estimated parametric distribution purses the nonparametric bootstrap, where samples are from $\hat{F}(t)$, the empirical cdf.

The power of the nonparametric bootstrap is that it does not require distributional assumptions, so many bootstrappers prefer the nonparametric approach.

Example, parametric BS of CV Suppose in the CV problem we assume

$$X_1, X_2, \ldots, X_n \overset{\text{iid}}{\sim} \text{Normal}(\mu, \sigma^2)$$

where $\underset{\sim}{\tau} = (\mu, \sigma^2$ is unknown. Here $n = 20$. To implement the parametric bootstrap assessment of the distributions of $\widehat{\text{CV}}$ and $\log \widehat{\text{CV}}$ we

1. estimate μ and σ^2 by MLE from data:
   $\hat{\mu} = \bar{x} = 4.303$

$$\hat{\sigma} = \sqrt{\tfrac{1}{n}\sum_i (x_i - \bar{x})^2} = 2.116$$

2. generate B bootstrap samples

$$\underset{\sim}{x}_i^* = \{x_{i1}^*, x_{i2}^*, \ldots, x_{in}^*\} \overset{\text{iid}}{\sim} \text{Normal}(\hat{\mu}, \hat{\sigma}^2)$$

   and from each compute $\widehat{\text{CV}}_i^*$ and $log(\widehat{\text{CV}}_i^*)$.

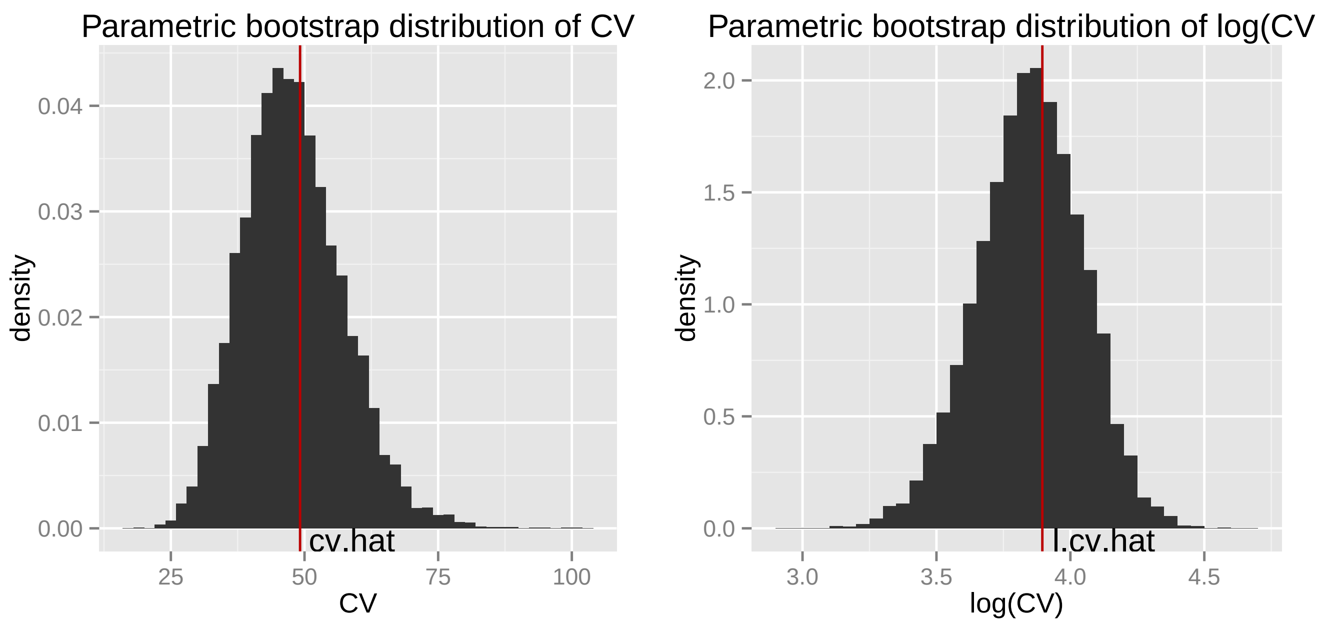
Note we I draw all the bootstrap samples with one call to `rnorm()`.

If the normal model is correct, then the parametric and non-parametric bootstraps are both estimating the sampling distribution of the estimated CV, and log(CV). The histograms from the two methods are fairly similar, although the parametric bootstrap distribution of the log(CV) appears to more symmetric.

```r
# Parametric bootstrap
R <- 1e4
# draw a matrix of samples
x.ast <- matrix(rnorm(R*n, mean = mu.hat, sd = sd.mle), nrow = R)
# row sd and mean give a vector of CVs
cv.bs <- 100 * n1.n * apply(x.ast, 1, sd) / apply(x.ast, 1, mean)
l.cv.bs <- log(cv.bs) # log CV

# bs summaries in data.frame
bs.sum <- data.frame(cv.bs, l.cv.bs)
```

```r
# CV
library(ggplot2)
p <- ggplot(bs.sum, aes(x = cv.bs))
p <- p + geom_histogram(aes(y = ..density..), binwidth=2)
p <- p + labs(title = "Parametric bootstrap distribution of CV")
p <- p + geom_vline(aes(xintercept=cv.hat), colour="#BB0000", linetype="solid")
p <- p + geom_text(data = data.frame(NA)
                   , aes(label = "cv.hat", x=cv.hat, y=0, hjust=-0.1, vjust=1))
p <- p + xlab("CV")
print(p)
```

```
# log(CV)
library(ggplot2)
p <- ggplot(bs.sum, aes(x = l.cv.bs))
p <- p + geom_histogram(aes(y = ..density..), binwidth=.05)
p <- p + labs(title = "Parametric bootstrap distribution of log(CV)")
p <- p + geom_vline(aes(xintercept=l.cv.hat), colour="#BB0000", linetype="solid")
p <- p + geom_text(data = data.frame(NA)
                   , aes(label = "l.cv.hat", x=l.cv.hat, y=0, hjust=-0.1, vjust=1))
p <- p + xlab("log(CV)")
print(p)
```


Parametric bootstrap distribution of CV
density
cv.hat
CV
Parametric bootstrap distribution of log(CV
density
l.cv.hat
log(CV)