Resampling

Contents

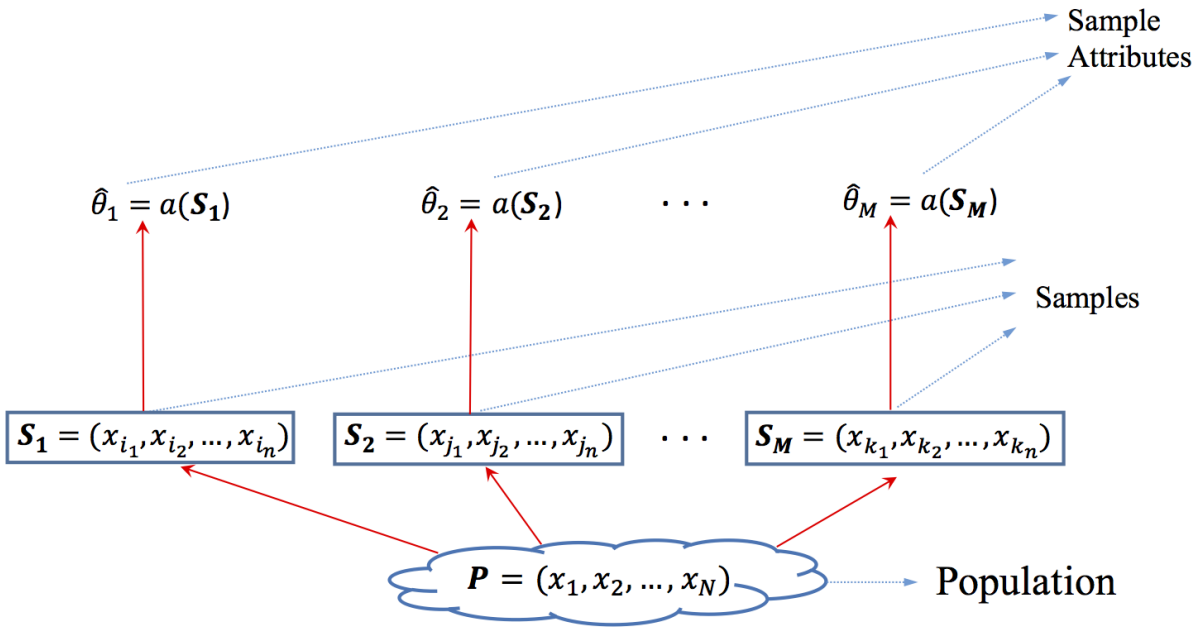
4.4 Resampling

4.4.0 Resampling Intuition

- As shown in previous sections, understanding the sampling behaviour of sample attributes is essential for making inferences about any population attribute. For example:
  - Knowing the sampling distribution of a discrepancy measure allows us to test hypotheses
  - Knowing the sampling distribution of a pivotal quantity allows us to construct confidence intervals

- To perform inference, we draw a sample $\mathcal{S}$ of size n from a study population $\mathcal{P}$ according to some sampling mechanism
  - then calculate the sample attribute $a(\mathcal{S})$ to estimate its population counterpart $a(\mathcal{P})$.

- To understand the sampling distribution of the attribute $a(\mathcal{S})$:
  - we draw M samples $\mathcal{S}_1, \ldots, \mathcal{S}_M$ and
  - use the values $a(\mathcal{S}_1), \ldots, a(\mathcal{S}_M)$ to inform us about the sampling distribution of $a(\mathcal{S})$.



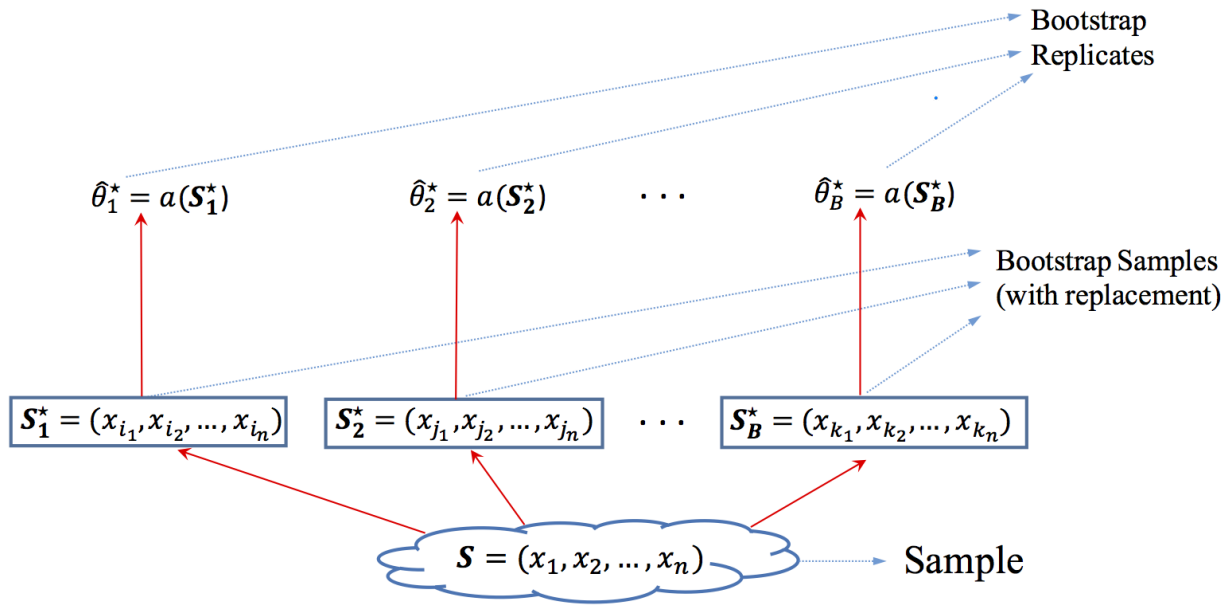
- However, this process requires undertaking repeated sampling from the population

- But in practice, the population from which our sample was taken cannot generally be repeatedly sampled
  - Therefore: we have only one sample.

- **Resampling** methods aim to *mimic* this process by repeatedly sampling $\mathcal{S}$ as if it were $\mathcal{P}$
  - In particular, we draw B samples $\mathcal{S}_1^\star, \ldots, \mathcal{S}_B^\star$ of size n independently from a population $\mathcal{P}^\star$.
  - Ideally, $\mathcal{P}^\star$ would be the study population $\mathcal{P}$, but as already mentioned this would require repeated sampling from the population
  - Instead we use our a sample $\mathcal{S}$ as an estimate of the population $\mathcal{P}$, i.e., $\widehat{\mathcal{P}} = \mathcal{S}$, or in usual *bootstrap notation* $\mathcal{P}^\star = \mathcal{S}$
    - Note, however, that in principle we could use any estimate of the study population.

- The sample population has only n units, so the without-replacement sampling mechanism will immediately exhaust the population. Therefore, we sample **with replacement**.

- Thus an approximate sampling distribution is obtained by drawing B samples $\mathcal{S}_1^\star, \ldots, \mathcal{S}_B^\star$ of size n from $\mathcal{P}^\star$ with replacement
  - and on each **bootstrap sample** calculate the attribute value: $a(\mathcal{S}_1^\star), \ldots, a(\mathcal{S}_B^\star)$



* the Bootstrap is just one method of resampling. Another commonly used method (and which was the precurser to the bootstrap) is called the jack-knife

4.4.1 The Bootstrap Method

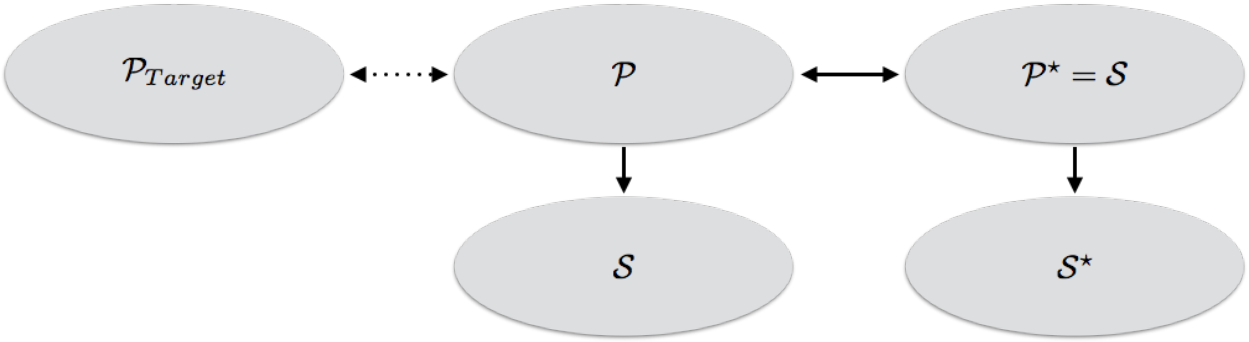
- The distribution of any attribute over the bootstrap samples $\mathcal{S}_i^\star$ from $\mathcal{P}^\star$ is a **bootstrap estimate** of the distribution of the same attribute over all possible samples $\mathcal{S}_i$ from $\mathcal{P}$

- This approach to mimicking the sampling distribution was named the **bootstrap** method when it was first proposed in 1979 by Bradley Efron.

- The word "bootstrap" conveys the notion of starting something up from nothing as in "pulling oneself over a fence by one's bootstraps".
  - It suggests something for nothing, or something impossible to achieve.

- In Efron (1979), other possible names were *Swiss Army Knife*, *Meat Axe*, *Swan-Dive*, *Jack-Rabbit*.

- The Bootstrap Path of Inductive Inference



Why is bootstrapping so important? What does it provide?

With a single sample, we are now able to construct an estimate of the sampling distribution of an attribute that does not rely on any assumptions about the form of that attribute.

Shark Length Example: Average

- Throughout this section we will illustrate relevant concepts using the population of shark encounters ($N = 65$) found in the `sharks` data

- The code below constructs an approximate sampling distribution for estimates of the average shark length based on samples of size $n = 6$
  - Note that there are $\binom{N}{n} = \binom{65}{6} = 82,598,880$ possible samples. Our approximate sampling distribution is based on $M = 10,000$ of these

```
sharks <- read.csv("/Users/nstevens/Dropbox/Teaching/STAT_341/Lectures/Data/sharks.csv")
popSharks <- rownames(sharks)

# A function to put n or N in the denominator of SD
# rather than n-1 or N-1
sdn <- function( y.pop ) {
  N = length(y.pop)
  sqrt( var(y.pop)*(N-1)/(N)  )    }

M <- 10^4
n = 6
set.seed(341)
samples <- sapply(1:M, FUN =function(m)  sample(popSharks, n, replace = TRUE) )
```

```
avePop <- mean(sharks[, "Length"])
avesSamp <- apply(samples, MARGIN = 2,
                  FUN = function(s){mean(sharks[s,"Length"])})
sampleErrors <- avesSamp - avePop

tmpAve <- mean(avesSamp)
tmpSD <- sdn(avesSamp)

sdsSamp <- apply(samples, MARGIN = 2,
                  FUN = function(s){sdn(sharks[s,"Length"])})
```

- With a single sample of $n = 6$ and we are interested in estimating the average shark length.
  - Should we expect the bootstrap to work if n is small?
  - The population has only $N = 65$ units so the sample size is a little under 10 percent of the population.

In general the effectiveness of the bootstrap relies on the assumption that S is a good approximation of P. With small n (versus large n) it is more likely that $S \neq P$

- We draw one sample $\mathcal{S}$ drawn from $\mathcal{P}$ using simple random sampling without replacement. The indices of the sampled rows are shown below.

```
popSize <- function(pop) {
  if (is.vector(pop))
  {if (is.logical(pop))
    ## then assume TRUE values identify units
    sum(pop) else length(pop)}
  else nrow(pop)
}

getSample <- function(pop, size, replace=FALSE) {
  N <- popSize(pop)
  pop[sample(1:N, size, replace = replace)]
}
```

```
set.seed(341)
n <- 6
S <- getSample(1:65, n, replace = FALSE)
S
```

```
## [1] 10 58 24  4 50 46
```

- Then we draw B bootstrap samples from this single sample.
  - There are $n^n = 6^6 = 46,656$ possible **bootstrap samples** of size $n = 6$ to select.
  - Here, we choose $B = 10,000$ bootstrap samples $S^\star_1, \ldots, S^\star_{10000}$:
  - We might choose a different B depending on the context.

```
Pstar <- S
B <- 10^4
set.seed(341)
Sstar <- sapply(1:B, FUN =function(b)  getSample(Pstar, n, replace = T))
```

- We then have a $6 \times B$ matrix of bootstrap sample indices, where each column represents a different sample

  - The first 10 bootstrap samples contain the units shown below

```
dim(Sstar)
```

```
## [1]     6 10000
```

```
Sstar[,1:10]
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
## [1,]   10   10   58   58   46   10   24    4   10     4
## [2,]   46   58   50   10   50    4   50   10   10    58
## [3,]   24   50   24   50   10   50   50   24    4    58
## [4,]   10   10   46    4   10   10    4   24    4    24
## [5,]   50   50   46   46   58   24   24   46   58    10
## [6,]   50   10    4    4   24   24   46   58   50    58
```

- We then compute whichever attribute might be of interest on each bootstrap sample.
  - Here we compute the average shark length for each bootstrap sample.

```
avesBootSamp <- sapply(1:B, FUN = function(i) mean(sharks[Sstar[,i], "Length"]))
```

- The initial ten bootstrap sample averages are

```
round(avesBootSamp[1:10],1)
```

```
##  [1] 112.3 128.7 121.8 122.0 121.0 100.5 108.8 110.8 120.3 142.5
```

- The collection of bootstrap averages is an estimate of the sampling distribution of the sample average
  - Note that it could also be thought of as a population of sorts and can thus be summarized like any other population

- Let us compare the **bootstrap estimate** of the sampling distribution to the approximation obtained by repeatedly sampling the population
  - The average shark lengths in the first 10 samples are:
- Histograms of both collections of sample estimates are shown below.

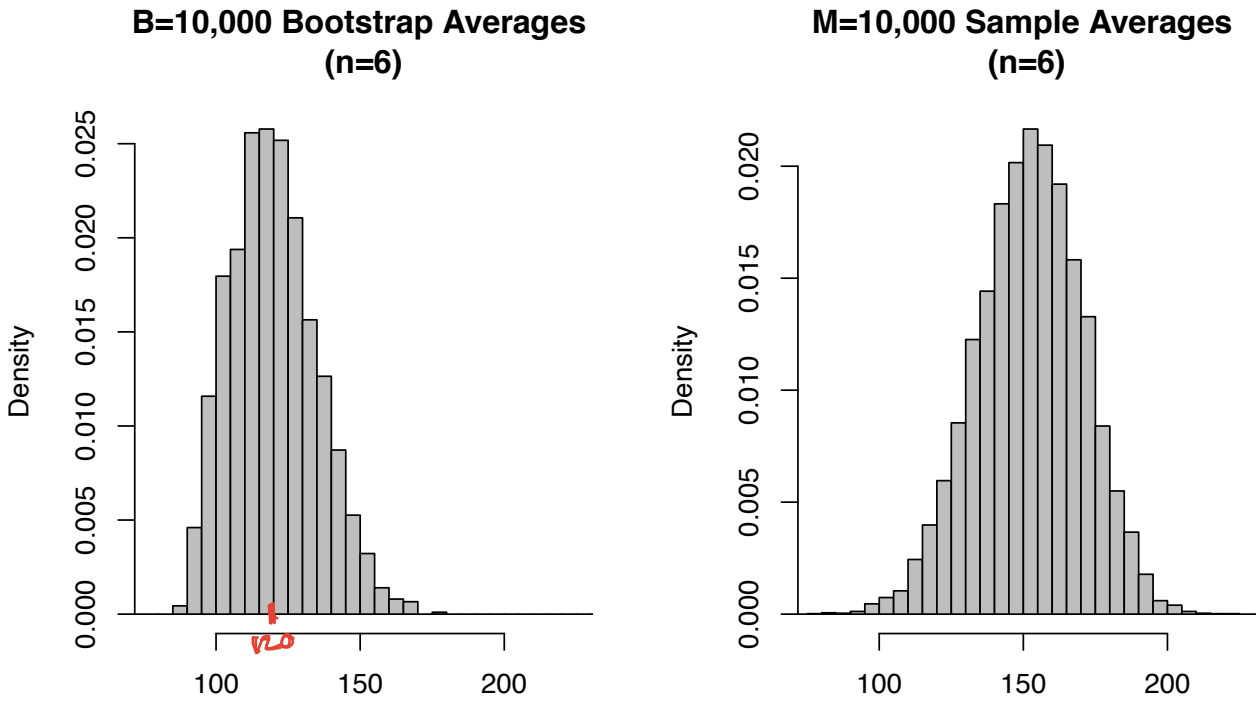
```
round(avesSamp[1:10],1)
```

```
##  [1] 141.3 154.0 161.3 125.2 145.3 151.5 143.7 147.2 156.2 134.5
```

```
savePar <- par(mfrow=c(1,2))

hPopAve <- hist(extendrange(c(avesSamp,avesBootSamp)), breaks = 50, plot = FALSE)
hist(avesBootSamp, xlim = range(avesSamp), breaks = hPopAve$breaks,
     freq = FALSE, col = "grey", main ="B=10,000 Bootstrap Averages \n(n=6)", xlab="")
hist(avesSamp, xlim = range(avesSamp), breaks = hPopAve$breaks,
     freq = FALSE, col = "grey", main ="M=10,000 Sample Averages \n(n=6)", xlab="")
```

*Two estimates of the sampling distribution

this one is expected to be closer to the truth



* Note that the bootstrap estimate is, at best, as good as $a(\mathcal{S})$ (Why?).

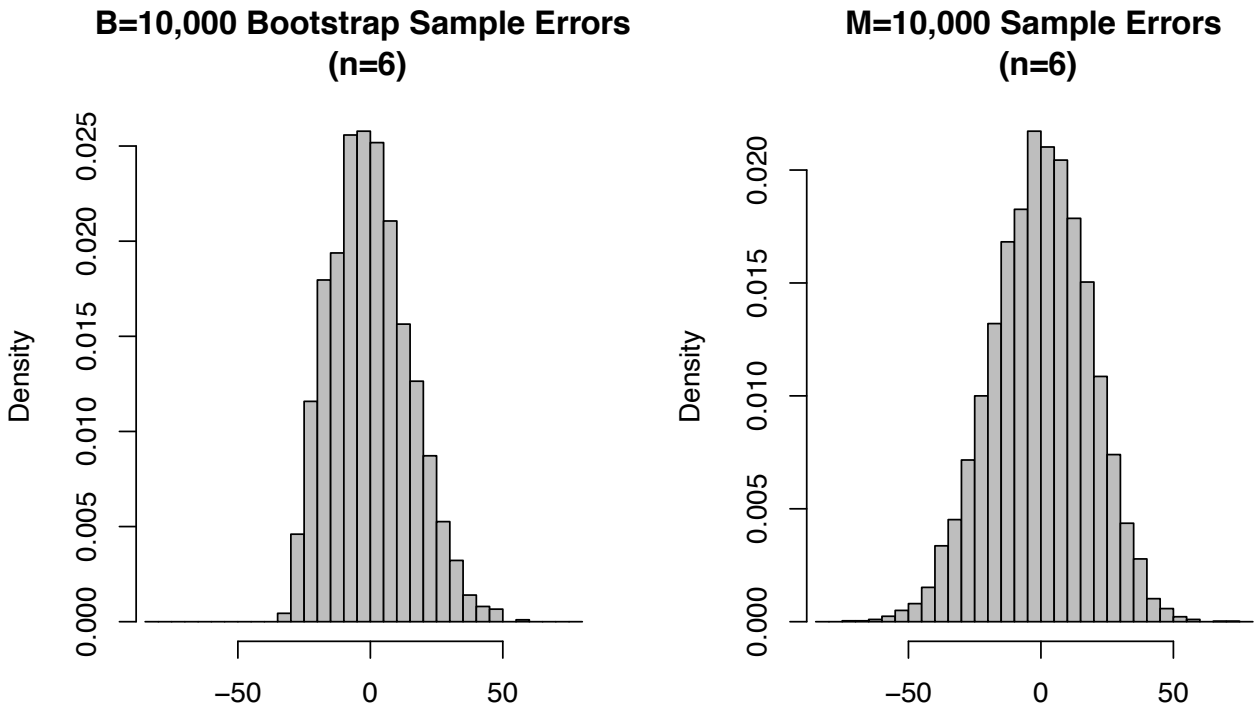
- As can been seen, the bootstrap distribution gives a sense of how an attribute varies.
  - * Thus, to get an idea of the variability in a sampling distribution we can take the standard deviation of the bootstrap distribution.

- To easily compare the variability we can construct histograms for each of the following errors:

$$\begin{aligned} \text{sample error} &= a(\mathcal{S}) - a(\mathcal{P}) \\ \text{bootstrap sample error} &= a(\mathcal{S}^\star) - a(\mathcal{S}) \end{aligned}$$

```
savePar <- par(mfrow=c(1,2))

range.avediff  <- extendrange(
  c(avesSamp- mean(avesSamp),
    avesBootSamp- mean(avesBootSamp)) )

hPopAvediff <- hist(range.avediff,
  breaks = 50, plot = FALSE)

hist(avesBootSamp - mean(avesBootSamp),
     xlim = range.avediff, breaks = hPopAvediff$breaks,
     freq = FALSE, col = "grey",
     main ="B=10,000 Bootstrap Sample Errors \n(n=6)", xlab="")
```

```r
hist(avesSamp - mean(avesSamp),
     xlim = range.avediff, breaks = hPopAvediff$breaks,
     freq = FALSE, col = "grey",
     main ="M=10,000 Sample Errors \n(n=6)", xlab="")
```



- The standard deviations in both of these distributions are as follows

```r
# Bootstrap Standard Deviation
sdn(avesBootSamp)
```

```
## [1] 14.82945
```

```r
# Repeated Sampling Standard Deviation
sdn(avesSamp)
```

```
## [1] 18.39039
```

Bootstrap Standard Deviation

- In general, for any attribute $a(\mathcal{P})$, the standard deviation of the corresponding sample attribute's estimator can be estimated from the bootstrap distributions with

$$\widehat{SD}_\star\left[\widetilde{a}(\mathcal{S}^\star)\right] = \sqrt{\frac{\sum_{b=1}^{B}\left(a(\mathcal{S}_b^\star) - \overline{a}^\star\right)^2}{B-1}}$$

  where $\overline{a}^\star = \frac{1}{B}\sum_{b=1}^{B} a(\mathcal{S}_b^\star)$ is the average of the attribute over the bootstrap samples $\mathcal{S}_1^\star, \ldots, \mathcal{S}_B^\star$.

- Since B is usually large, it does not make any practical difference whether we use B or $B-1$ in the denominator of the standard deviation.

- This is an estimate of the standard deviation of the sampling distribution for the attribute $a(\mathcal{S})$.

Special Case: Inference for a Population Average

- In the special case of the arithmetic average $a(\mathcal{S}) = \frac{1}{n}\sum_{u\in\mathcal{S}} y_u$ the bootstrap estimate of its standard deviation is

$$\widehat{SD}_\star[\overline{Y}] = \sqrt{\frac{\sum_{b=1}^{B}(\overline{y}_b^\star - \overline{y}^\star)^2}{B-1}}$$

  where $\overline{y}^\star = \frac{1}{B}\sum_{b=1}^{B}\overline{y}_b^\star$.
  - (as was done above in the `shark` example)

- But we also know that the standard deviation can be estimated with

$$\widehat{SD}[\overline{Y}] = \frac{\widehat{\sigma}}{\sqrt{n}}\sqrt{\frac{N-n}{N-1}} \quad \text{where} \quad \widehat{\sigma} = \sqrt{\frac{\sum_{u\in\mathcal{S}}(y_u-\overline{y})^2}{n}}$$

- Calculating these two estimates in the context of the `shark` example yields:

```
n=6
N=dim(sharks)[1]
c(sdn(avesBootSamp), sdn(sharks[S, "Length"])/sqrt(n)*sqrt((N-n)/(N-1)) )
```

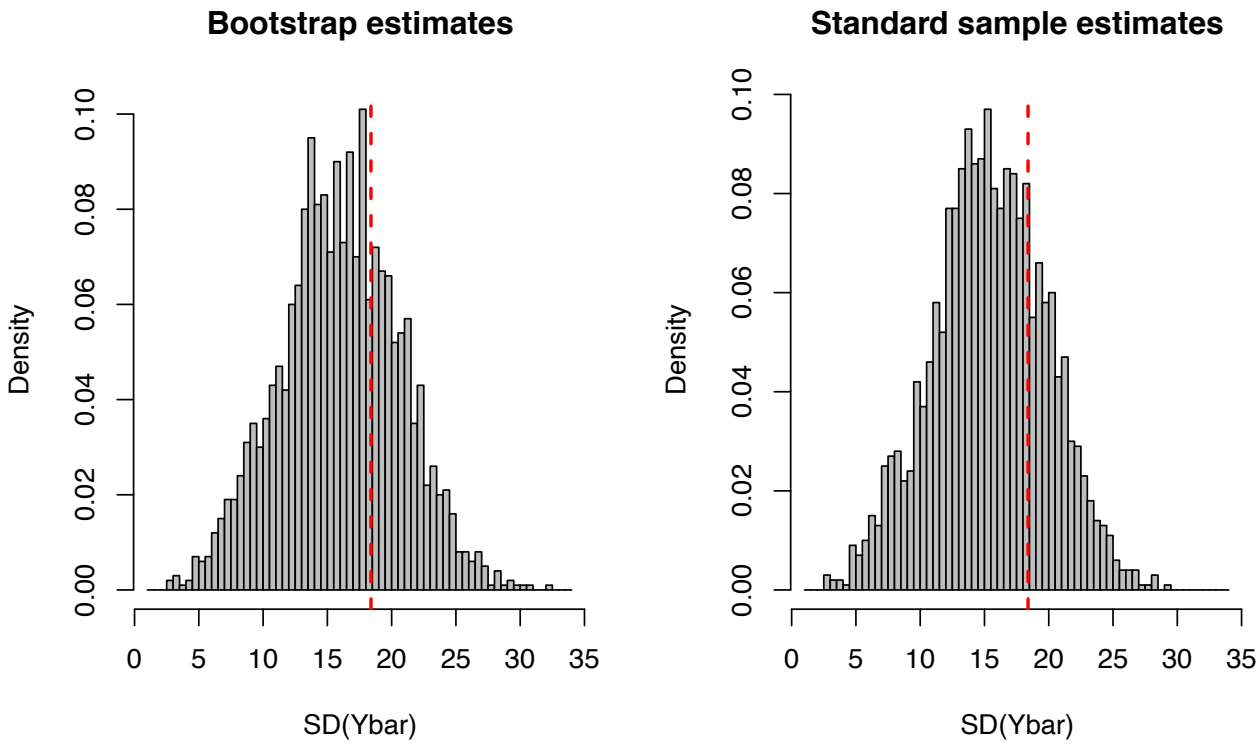
```
## [1] 14.82945 14.31729
```

- **Note:** if the sample is not a good representation of the population, these two numbers might be quite different.

- To more fully compare the two methods of estimating standard deviation we randomly select $M = 2000$ samples of size $n = 6$ from the population, and for each sample:
  - calculate $\frac{\widehat{\sigma}}{\sqrt{n}}\sqrt{\frac{N-n}{N-1}}$, yielding 2000 repeated-sample-based estimates of $\widehat{SD}[\overline{Y}]$.
  - and also generate $B = 200$ bootstrap samples used to obtain 2000 bootstrap estimates of $\widehat{SD}[\overline{Y}]$.

```
set.seed(341)
numSamps <-2000
n <- 6

sampSes <- sapply(1:numSamps, FUN = function(i) getSample(popSharks, n))
B <- 200

valsBoot = t(apply(sampSes,2, function(sam=NULL, B=NULL) {
  avesTemp = unlist(Map( function(i) {
```

```
      mean(sharks[sample(sam, n, replace = TRUE), "Length"]) }, 1:B) )

  c( sdn(avesTemp), sdn( sharks[sam, "Length"] )/sqrt(n)*sqrt((N-n)/(N-1)))
  }, B=B ))



colnames(valsBoot) <- c("sdBoot", "sdStandard")

par(mfrow=c(1,2))
hTmp <- hist(extendrange(valsBoot), breaks = 50, plot = FALSE)
hist(valsBoot[,"sdBoot"], xlim = extendrange(valsBoot),
     breaks = hTmp$breaks, freq = FALSE, col = "grey",
     main ="Bootstrap estimates", xlab="SD(Ybar)")
abline(v=sdn(avesSamp), col="red", lty=2, lwd=2)

hist(valsBoot[,"sdStandard"], xlim =extendrange(valsBoot),
     breaks = hTmp$breaks, freq = FALSE, col = "grey",
     main ="Standard sample estimates", xlab="SD(Ybar)")
abline(v=sdn(avesSamp), col="red", lty=2, lwd=2)
```

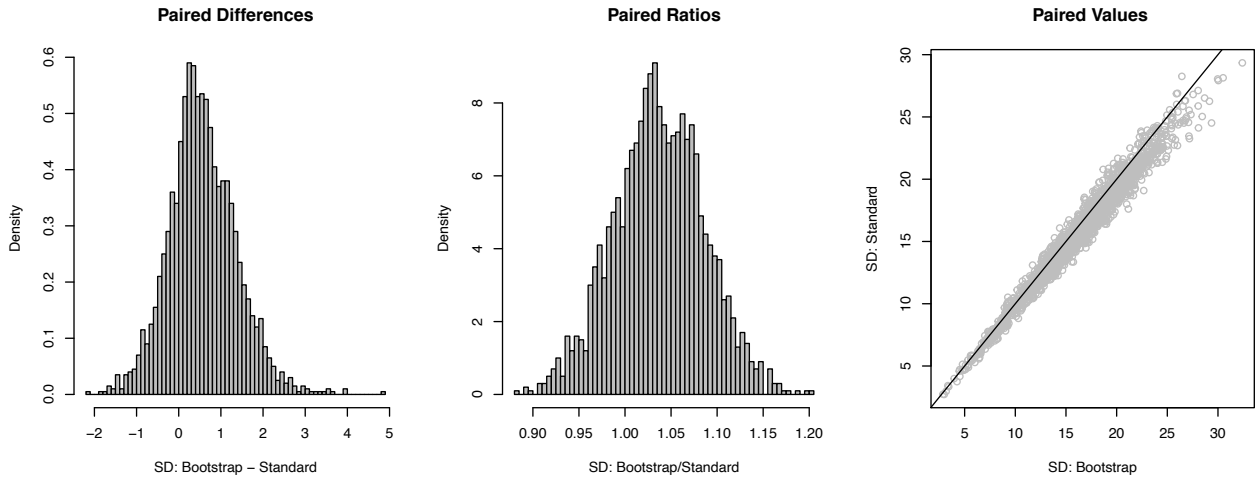


- The two histograms are not too dissimilar in shape, but the bootstrap estimator of standard deviation has produced, on average, slightly larger values than the standard approach.

- This can be more easily seen by constructing plots of the pairs of 2000 standard deviation estimates:

```
par(mfrow=c(1,3))
hist(valsBoot[,"sdBoot"] - valsBoot[,"sdStandard"], main = "Paired Differences", breaks=50, col="grey",
```

```
hist(valsBoot[,"sdBoot"] / valsBoot[,"sdStandard"], main = "Paired Ratios", breaks=50, col="grey", freq=
plot(valsBoot[,"sdBoot"], valsBoot[,"sdStandard"], main = "Paired Values ",col="grey", xlab="SD: Bootst
abline(a=0,b=1)
```



A Comment on n Versus $n-1$

- In the calculations above the function `sdn(...)` was introduced and used which is an implementation of

$$\widehat{\sigma} = \sqrt{\frac{\sum_{u \in \mathcal{S}} (y_u - \overline{y})^2}{n}}$$

  which has n as a divisor. The built-in function `sd(...)` in R is an implementation of

$$\widehat{\sigma} = \sqrt{\frac{\sum_{u \in \mathcal{S}} (y_u - \overline{y})^2}{n-1}}$$

- For bootstrap interval calculations, a divisor of n is preferred (since we are treating the sample as a population)
  - This version has the advantage of being **replication invariant**.
  - Replication invariant estimates are preferred and are often called **plug in estimates** in the bootstrap literature e.g. see Efron and Tibshirani (1994) .
- However, when n is reasonably large, there will be little practical difference between the two.

Shark Length Example: Median, Standard Deviation and Skewness

- In the previous investigation we saw that the bootstrap approach to calculating standard deviations of sampling distributions closely matched the (computationally less intensive) approach that exploited some theory specific to the estimation of population averages.

- The huge advantage that the bootstrap approach has is that it is applicable to any sample attribute not just the arithmetic average.

- In this section we also consider the
  - median
  - standard deviation
  - skewness

- Using the sample, 10, 58, 24, 4, 50, 46, and $B = 10,000$ boostrap samples we can construct bootstrap estimates of each attribute's sampling distribution and hence standard devation

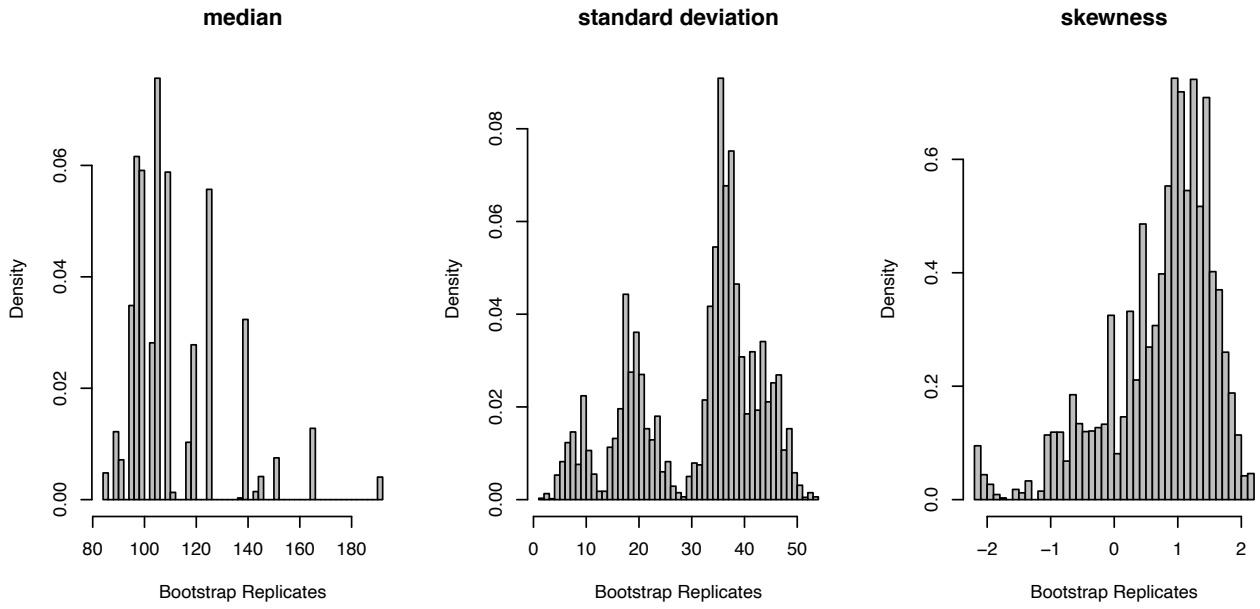
- The sample estimates of median, standard deviation, and skewness are respectively:

```
skew  <- function(z) { 3*(mean(z) - median(z))/sdn(z) }

sam.len = sharks[S, "Length"]
sam.est = c(median(sam.len), sdn(sam.len), skew(sam.len))
round(sam.est,2)
```

```
## [1] 105.00  36.53   1.25
```

- Generating $B = 10,000$ bootstrap samples and calculating each attribute on every sample yields the following estimated sample distributions:

```
bootAttr <- apply(Sstar, 2, function(sam, len) {
                pop = len[sam]
                c( median(pop), sdn(pop), skew(pop) )
                },
                len = sharks[, "Length"] )
par(mfrow=c(1,3))
namHist = c("median", "standard deviation", "skewness")

for (i in 1:3) hist(bootAttr[i, ], main = namHist[i], xlab="Bootstrap Replicates",
     breaks=50, col="grey", freq=FALSE)
```



- The corresponding bootstrap estimate of the standard deviation for each sampling distribution is:

```
round(apply(bootAttr,1,sd),3)
```

```
## [1] 18.370 11.643  0.825
```

So What are Bootstrap Distributions Good For?

Estimating (and Correcting for) Sampling Bias

- Recall that the **sampling bias** of an attribute is defined as:

$$\text{Sampling Bias} = E[a(\mathcal{S})] - a(\mathcal{P})$$

- We can use the bootstrap to estimate sampling bias via

$$\widehat{\text{Sampling Bias}} = \text{average bootstrap sample error} = \overline{a}^\star - a\left(\mathcal{S}\right)$$

where $\overline{a}^\star = \frac{1}{B}\sum_{b=1}^{B} a(\mathcal{S}_b^\star)$ is the average of the attribute over the bootstrap samples $\mathcal{S}_1^\star, \ldots, \mathcal{S}_B^\star$.

- When an estimator is unbiased we would like to "correct" it
  - i.e. make a biased estimator unbiased.

- If, in theory, $a(\mathcal{S})$ was biased and we knew the bias then we could subtract the correction from our attribute to make a new attribute $a^\star(\mathcal{S})$ that is unbiased:

$$a^\star(\mathcal{S}) = a(\mathcal{S}) - \text{Sampling Bias}$$

  - To verify that $a^\star(\mathcal{S})$ that is unbiased:

$$E[a^\star(\mathcal{S})] = E[a(\mathcal{S}) - \text{Sampling Bias}] = E[a(\mathcal{S}) - E[a(\mathcal{S})] + a(\mathcal{P})] = a(\mathcal{P})$$

- However, we don't typically *know* the sampling bias, so instead we could use the bootstrap estimate of it.
  - The bias-corrected bootstrap estimate is

$$a(\mathcal{S}) - \widehat{\text{Sampling Bias}}\,[a(\mathcal{S})] = a(\mathcal{S}) - [\overline{a}^\star - a(\mathcal{S})] = 2a(\mathcal{S}) - \overline{a}^\star$$

Confidence Intervals & Hypothesis Tests

- As was demonstrated in previous sections of the notes, the sampling distribution of an attribute was integral in calculating confidence intervals and performaing hypothesis tests
- Approximate confidence intervals and hypothesis tests can now also be based on the bootstrap estimate of a sampling distribution.
- We explore bootstrap-based confidence in the next section of the notes.
- Bootstrap-based hypothesis tests are not covered in this offering of STAT 341.

4.4.2 Bootstrap Confidence Intervals

- The bootstrap distribution provides a proxy for the sampling distribution for any sample attribute $a(\mathcal{S})$.
  - So we can use it to construct (at least approximate) confidence intervals for the unknown population attribute $a(\mathcal{P})$.

Naive Normal-Theory Intervals

- Recall that confidence intervals for sample averages tend to have the following structure:

$$\left[\overline{y} - c\widehat{SD}(\overline{Y}),\ \ \overline{y} + c\widehat{SD}(\overline{Y})\right]$$

- Under an assumption of normality we might pick c such that $P(Z \leq c) = 1 - p/2$ generates a $100(1-p)\%$ confidence interval

- If the bootstrap distribution is approximately normal, rather than estimating $\widehat{SD}(\overline{Y})$ by $\frac{\widehat{\sigma}}{\sqrt{n}}\sqrt{\frac{N-n}{N-1}}$
  - we might estimate $\widehat{SD}(\overline{Y})$ using the standard deviation of the bootstrap distribution of $\overline{Y}$.
  - The attraction of this approach, if it works, is that the same approach could be used for any attribute $a(\mathcal{S})$.

- A 95% **normal bootstrap interval** for a population attribute $a(\mathcal{P})$ is

$$a(\mathcal{S}) \pm 1.96\widehat{SD}_\star\left[a(\mathcal{S})\right]$$

  where $\widehat{SD}_\star$ is the bootstrap estimate of the standard deviation.

- **Question:** What does 95% mean here?

We expect 95% of the bootstrap samples to contain a(P)

Example: Average Shark Length

- A 95% normal bootstrap confidence interval for the average shark length is:

```
mean(sharks[S, "Length"]) + 1.96*c(-1,1)*sdn(avesBootSamp)
```

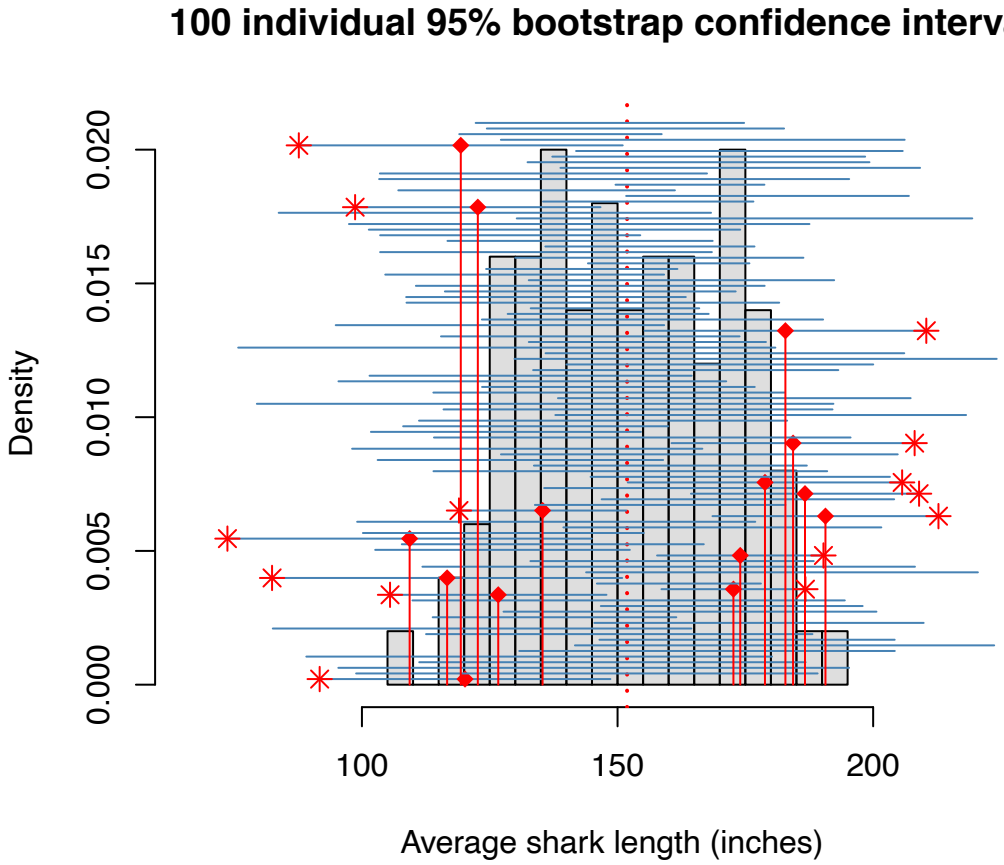
```
## [1]  91.10095 149.23239
```

- Note, the population average is 151.86

Estimating Coverage Probability: The Normal Bootstrap Interval

- If we use the normal bootstrap interval then we should investigate its properties such as determining if we obtain the proper coverage probability.
- The average shark length of great whites was 151.86 inches (in our population of $N = 65$ encounters).
- Suppose we are interested in a 95% confidence interval for the population mean.
- To get an estimate of the coverage probability for the bootstrap confidence interval, we generate 100 samples of size $n = 6$ and for each sample:
  - generate $B = 1000$ bootstrap samples to obtain the bootstrap estimate of the standard deviation, $\widehat{SD}(\overline{Y})$
  - and then construct the 95% confidence interval (using $c = 1.96$)

-These 100 intervals (and their coverage) are illustrated below



- As can be seen, only 86 of the 100 bootstrap intervals actually cover the average in the population.
  - This is a much lower coverage proportion than the expected 95% suggested by using $c = 1.96$.

- Under-coverage is perhaps not unexpected as the c value is based on a normal distribution which may not apply.

Bootstrap-t Confidence Intervals

Let's think of this as approximately pivotal for any attribute $a(\mathcal{P})$

- We have seen that the quantity

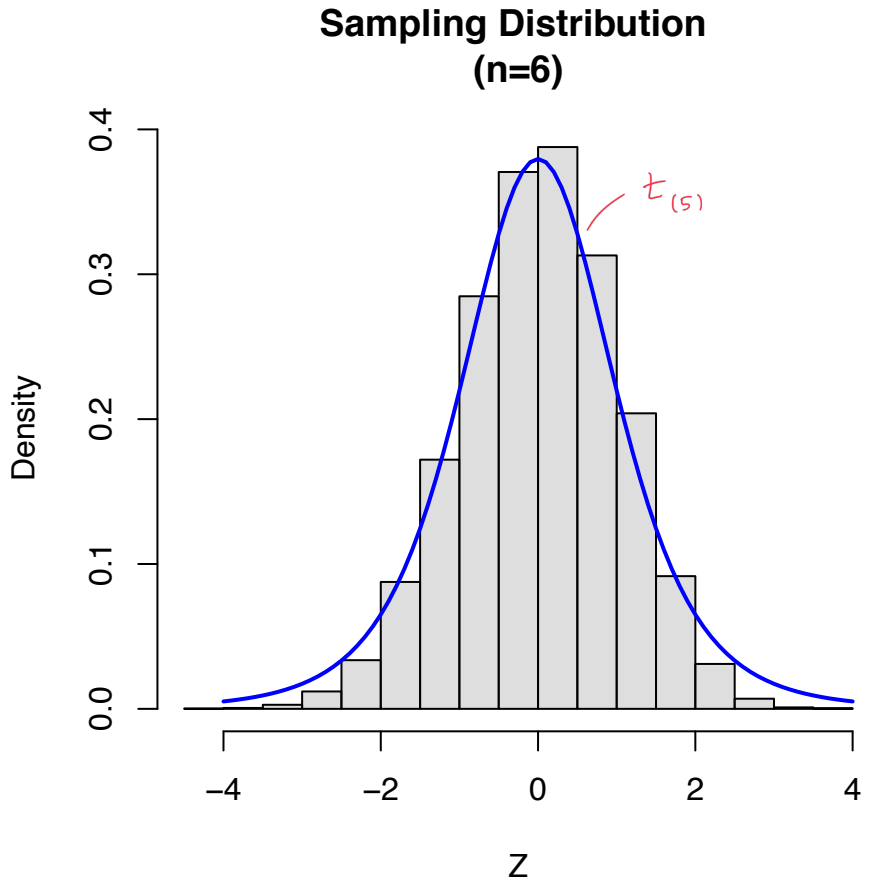
$$Z = \frac{\widetilde{a}(\mathcal{S}) - a(\mathcal{P})}{\widetilde{SD}[\widetilde{a}(\mathcal{S})]}$$

  is pivotal for $a(\mathcal{S}) = \overline{y}$
  - Its sampling distribution (over all possible samples) is well approximated by a t-density.

- For example when $a(\mathcal{P})$ is average shark length, the sampling distribution (with $n = 6$) is shown below.
  - Overlaid on this histogram is the $t_{(5)}$ density

```
ZPop <- (avesSamp - mean(sharks[,"Length"]))/sdn(avesSamp)
hist(ZPop, freq = FALSE, breaks = 20, col = adjustcolor("grey", 0.5),
```

```
     main = "Sampling Distribution \n(n=6)",
     xlab = "Z")

lines(x = seq(-4,4,0.1), y = dt(x = seq(-4,4,0.1), df = 5), col = "blue", lwd = 2)
```



- This suggests that instead of determining c from a normal distribution, we could choose c from a t-distribution with $n-1$ degrees of freedom in the "naive normal-theory intervals"
  - The t-distribution approximation of a sampling distribution may work for certain attributes, but not all of them.
  - This requires $\widetilde{a}(\mathcal{S})$ to be approximately normal over all possible samples. (When might this be true?)
  - If $a(\mathcal{P})$ is the median or a measure of skewness, we would not expect the t-distribution to be a good approximation.

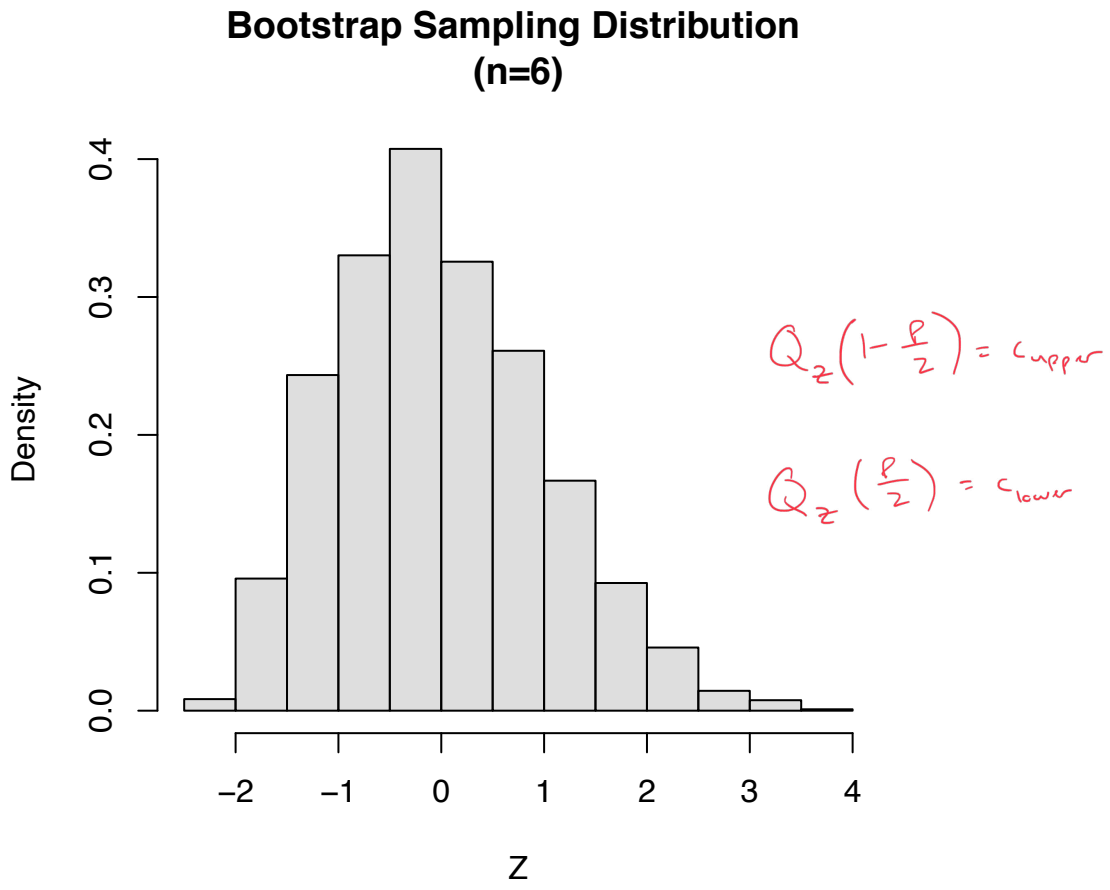
- We could consider taking this approach (i.e., calculating Z as above) for any attribute $a(\mathcal{P})$ – not just the average – but the t-distribution approximation should probably be avoided

- Instead of approximating the sampling distribution with a t-distribution, we could use quantiles from the actual sampling distribution to determine appropriate c values.

- We have seen that the bootstrap may be used to approximate the sampling distribution of $a(\mathcal{S})$
  - Here we will use the bootstrap to estimate the sampling distribution of
$$Z = \frac{\widetilde{a}(\mathcal{S}) - a(\mathcal{P})}{\widetilde{SD}[\widetilde{a}(\mathcal{S})]}$$
    and use it to construct confidence intervals.
  - On the negative side: this requires computation.
  - On the positive side: the bootstrap automatically adjusts its shape (and hence quantiles, etc.) to the form of $\widetilde{a}(\mathcal{S})$.

- To use the bootstrap to approximate the sampling distrbution we must calculate
$$Z = \frac{\widetilde{a}(\mathcal{S}) - a(\mathcal{P})}{\widetilde{SD}[\widetilde{a}(\mathcal{S})]}$$
  differently than we did before. In particular, we replace:
  - the sample $\mathcal{S}$ with the bootstrap sample $\mathcal{S}^\star$, and
  - the population $\mathcal{P}$ with the estimate $\mathcal{P}^\star$ (the sample $\mathcal{S}$)
$$Z^\star = \frac{\widetilde{a}(\mathcal{S}^\star) - a(\mathcal{P}^\star)}{\widetilde{SD}\widetilde{a}(\mathcal{S}^\star)]} = \frac{\widetilde{a}(\mathcal{S}^\star) - a(\mathcal{S})}{\widetilde{SD}[\widetilde{a}(\mathcal{S}^\star)]}$$

```
ZBoot <- (avesBootSamp - mean(sharks[Pstar,"Length"]))/sdn(avesBootSamp)
hist(ZBoot, freq = FALSE, breaks = 20, col = adjustcolor("grey", 0.5),
     main = "Bootstrap Sampling Distribution \n(n=6)",
     xlab = "Z")
```

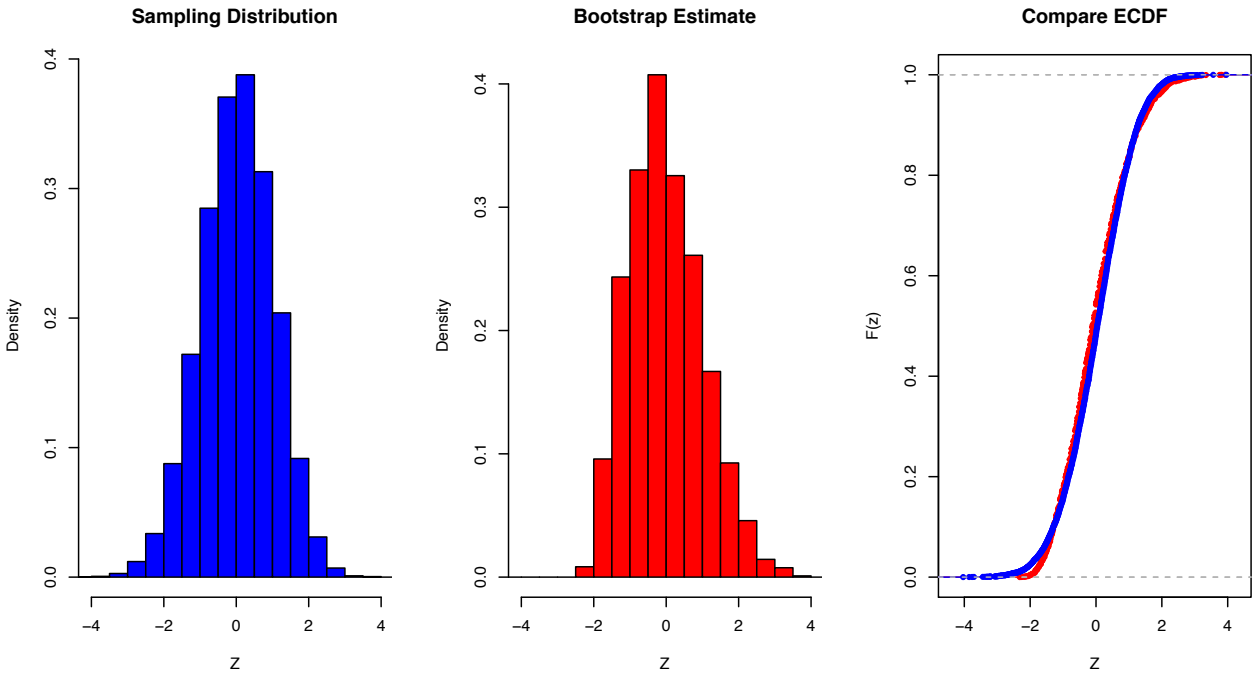


- We use the bootstrap distribution of Z to find values c_{lower} and c_{upper} such that

$$Pr(c_{lower} \leq Z \leq c_{upper}) = (1-p)$$

with $(1-p)$ being the intended coverage probability.

- The plots below compare these two approaches:



- The values of c obtained by each method are: (for a 95% CI)
  - t-distribution: $c_{upper} = Q_z(0.975) = 2.57$ and $c_{lower} = Q_z(0.025) = -2.57$
  - Bootstrap distribution: $c_{upper} = Q_z(0.975) = 2.14$ and $c_{lower} = Q_z(0.025) = -1.71$

This approach requires no distributional assumptions.

The General Approach

- For a given sample $\mathcal{S}$
  - Calculate $a(\mathcal{S})$
  - Using B bootstrap samples $\mathcal{S}_1^\star, \ldots, \mathcal{S}_B^\star$ from $\mathcal{S}$ calculate $\widehat{SD}_\star\left[a(\mathcal{S})\right]$

- For each of the B bootstrap samples from above:
  - Calculate $a(\mathcal{S}_b^\star)$ and $\widehat{SD}\left[a(\mathcal{S}_b^\star)\right]$ and then

$$z_b = \frac{a(\mathcal{S}_b^\star) - a(\mathcal{S})}{\widehat{SD}\left[a(\mathcal{S}_b^\star)\right]}$$

- From the values $z_1, \ldots, z_B$, find
  - $c_{lower} = Q_z\left(p/2\right)$ and $c_{upper} = Q_z\left(1 - p/2\right)$

- Then a $100(1-p)\%$ bootstrap-t confidence interval is

$$\left[a(\mathcal{S}) - c_{upper} \times \widehat{SD}_\star\left[a(\mathcal{S})\right], a(\mathcal{S}) - c_{lower} \times \widehat{SD}_\star\left[a(\mathcal{S})\right]\right]$$

Note the signs and positions of the constants c_{lower} and c_{upper} in the interval definition.

- This method (so far) requires an analytic form to calculate $\widehat{SD}\left[a(\mathcal{S}_b^\star)\right]$ (i.e, the standard deviation of the estimator given a single sample)
  - This may or may not be available

Example: Average Shark Length

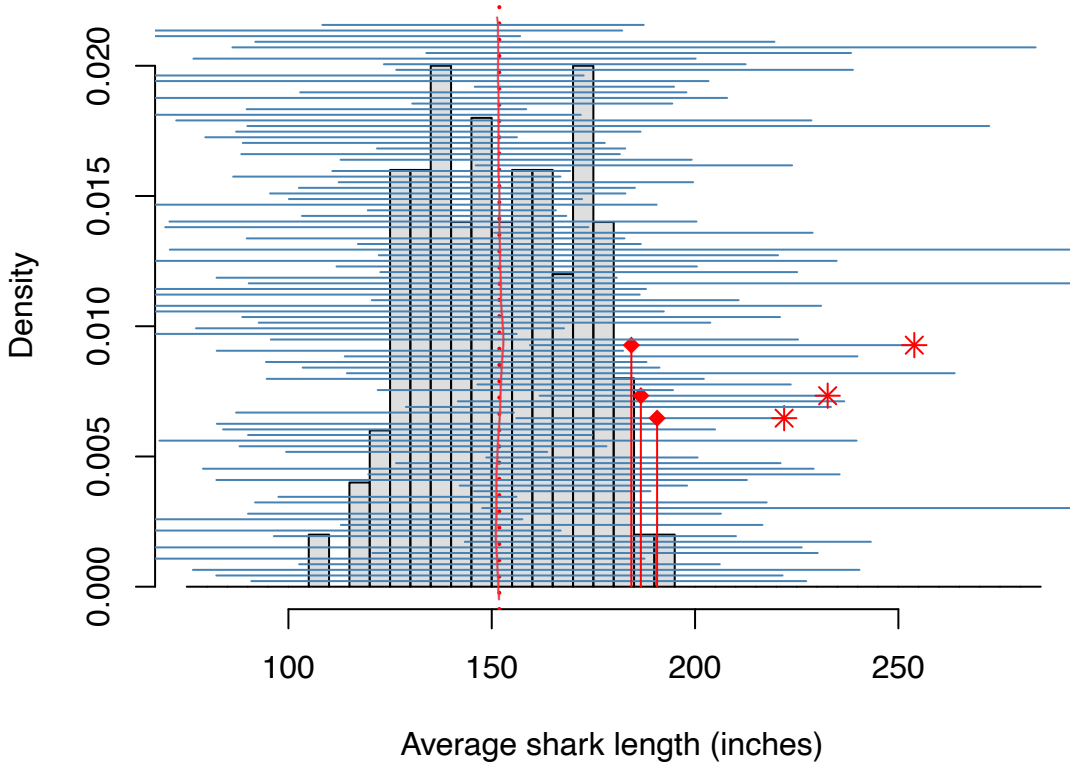
- The Bootstrap-t confidence interval for the average shark length is:

```
as.numeric(c(mean(sharks[S, "Length"]) - quantile(ZBoot, 0.975)*sdn(avesBootSamp),
  mean(sharks[S, "Length"]) - quantile(ZBoot, 0.025)*sdn(avesBootSamp)))
```

```
## [1]  88.5 145.5
```

Estimating Coverage Probability: Bootstrap-t Intervals

- Here we investigate the coverage of confidence intervals calculated in this way



- We see that 97 of the 100 bootstrap intervals cover the average in the population.

  In general, we would expect these intervals to be wider due to the bootstrap estimates of c_{lower} and c_{upper} than earlier intervals.

The Double Bootstrap

- When $a(\mathcal{S}) = \overline{y}$ we have an analytic form for its standard deviation:

$$SD[\overline{Y}] = \frac{\sigma}{\sqrt{n}} \times \sqrt{\frac{N-n}{N-1}}$$

  where replacing σ by $\widehat{\sigma}$ gives an estimate $\widehat{SD}[\overline{Y}]$ based on the sample values y_u for $u \in \mathcal{S}$.

More generally, when $a(\mathcal{S})$ is a Horvitz-Thompson estimate we also have an analytic form for $\widehat{SD}[a(\mathcal{S})]$

- However, very often an analytic solution is not available for $\widehat{SD}[a(\mathcal{S})]$

– In which case an estimate can be obtained by using the bootstrap:

$$\widehat{SD}_{\star}\left[\widetilde{a}(\mathcal{S})\right] = \sqrt{\frac{\sum_{b=1}^{B}\left(a(\mathcal{S}_b^{\star}) - \overline{a}^{\star}\right)^2}{B-1}}$$

where $\overline{a}^{\star} = \frac{1}{B}\sum_{b=1}^{B} a(\mathcal{S}_b^{\star})$

- But in the bootstrap-t, we need an estimate of $SD[a(\mathcal{S}_b^{\star})]$ *for each* bootstrap sample $\mathcal{S}_b^{\star}$!

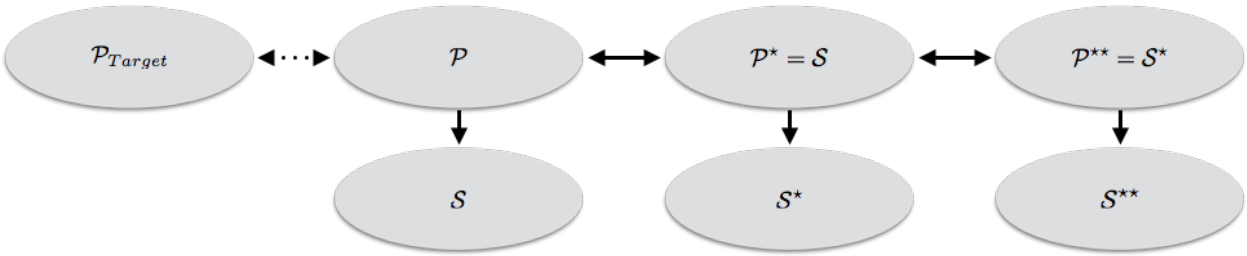
- In order to determine $\widehat{SD}[a(\mathcal{S}_b^{\star})]$ for each bootstrap sample $\mathcal{S}_b^{\star}$ we use the **double bootstrap**
  - Here we apply the bootstrap method to each bootstrap sample $\mathcal{S}_b^{\star}$.

- To apply a **bootstrap within a bootstrap** for each bootstrap sample $\mathcal{S}_b^{\star}$
  - We generate D bootstrap samples, $\mathcal{S}_1^{\star\star}, \ldots, \mathcal{S}_D^{\star\star}$, each with replacement from a population now defined as $\mathcal{P}^{\star\star} = \mathcal{S}_b^{\star}$.
  - The standard deviation of the corresponding values $a(\mathcal{S}_1^{\star\star}), \ldots, a(\mathcal{S}_D^{\star\star})$ will provide the estimate $\widehat{SD}[a(\mathcal{S}_b^{\star})]$

- This estimate $\widehat{SD}[a(\mathcal{S}_b^{\star})]$ is then substituted into **The General Approach** for bootstrap-t confidence intervals

$$Z_b = \frac{a(\mathcal{S}_b^{\star}) - a(\mathcal{S})}{\widehat{SD}[a(\mathcal{S}_b^{\star})]}$$

- Just as with the original bootstrap, this additional bootstrap adds another link in the chain of the inductive path as shown below.



Example: Average Shark Length

- The `bootstrap_t_interval` function defined below is a general-purpose function that may used to calculate a bootstrap-t confidence interval for *any* attribtute
  - The function requires as input an attribute function `a` which will calculate $a(\mathcal{S})$ for any $\mathcal{S} \subset \mathcal{P}$.
  - In the case of the average shark length, this can be written as

```
a <- function(S) {mean(sharks[S, "Length"])}
```

- Here is the `bootstrap_t_interval` function:

```
bootstrap_t_interval <- function(S, a,
                                 confidence = 0.95,
                                 B = 1000, D = 30){
  ## Here S is the sample, a is a scalar-valued function a(S) of a sample S
  ## which returns the value for S of that attribute of interest
  ## confidence is the level of confidence
  ## B is the outer bootstrap count of replicates used to
  ## calculate the lower and upper limits
  ## D the inner bootstrap count of replicates used to
  ## estimate the standard deviation of the sample attribute
  ## for each (outer) bootstrap sample
  Pstar <- S
  aPstar <- a(Pstar)
  ## get (outer) bootstrap values
  bVals <- sapply(
    1:B,
    FUN = function(b) {
      Sstar <- getSample(Pstar, sampleSize, replace = TRUE)
      aSstar <- a(Sstar)
      ## get (inner) bootstrap values to
      ## estimate the SD
      Pstarstar <- Sstar
      SD_aSstar <- sdn(
        sapply(1:D,
               FUN = function(d){
                 Sstarstar <- getSample(Pstarstar, sampleSize, replace = TRUE)
                 ## return the attribute value
                 a(Sstarstar)
               }
        )
      )
      z <- (aSstar - aPstar)/SD_aSstar
      ## Return the two values
      c(aSstar = aSstar, z = z)
    })
  SDhat <- sdn(bVals["aSstar",])
  zVals <- bVals["z",]
  ## Now use these zVals to get the lower and upper
  ## c values.
  cValues <- quantile(zVals,
                      probs = c((1 - confidence)/2,  (confidence +1)/2),
                      na.rm = TRUE)
```

```
  cLower <- min(cValues)
  cUpper <- max(cValues)
  interval <- c(lower = aPstar - cUpper * SDhat,
                middle = aPstar,
                upper = aPstar - cLower  * SDhat)
  interval
}
```

- A 95% bootstrap-t confidence interval for the average shark length is

```
S <- getSample(popSharks, size = 6, replace = TRUE)
bootstrap_t_interval(S, a)
```

```
##     lower    middle     upper
##  75.13228 142.16667 192.44412
```

Estimating Coverage Probability: The Double Bootstrap

- Here we investigate the coverage of confidence intervals calculated in this way
  - Note that this simulation takes a decent amount of time to complete

```
confidence <- 0.95
B <- 1000
set.seed(341)

intervals <- sapply(1:numIntervals,
    FUN = function (i) {
    bootstrap_t_interval(samps[,i], a = a,
          confidence = confidence, B=B, D=30) } )
```

- We see that 96 of the 100 bootstrap intervals cover the population attribute.

Comments:

- According to [Efron and Tibshirani (1994), pp. 161-162] the bootstrap-t intervals are best suited for attributes which measure "location parameters" like the average, the median, a particular quantile, etc.

- For attributes that do not measure location, it may be necessary to first transform the attribute to a scale that produces bootstrap-t that have good coverage probabilities,
  - Or consider using the percentile method.

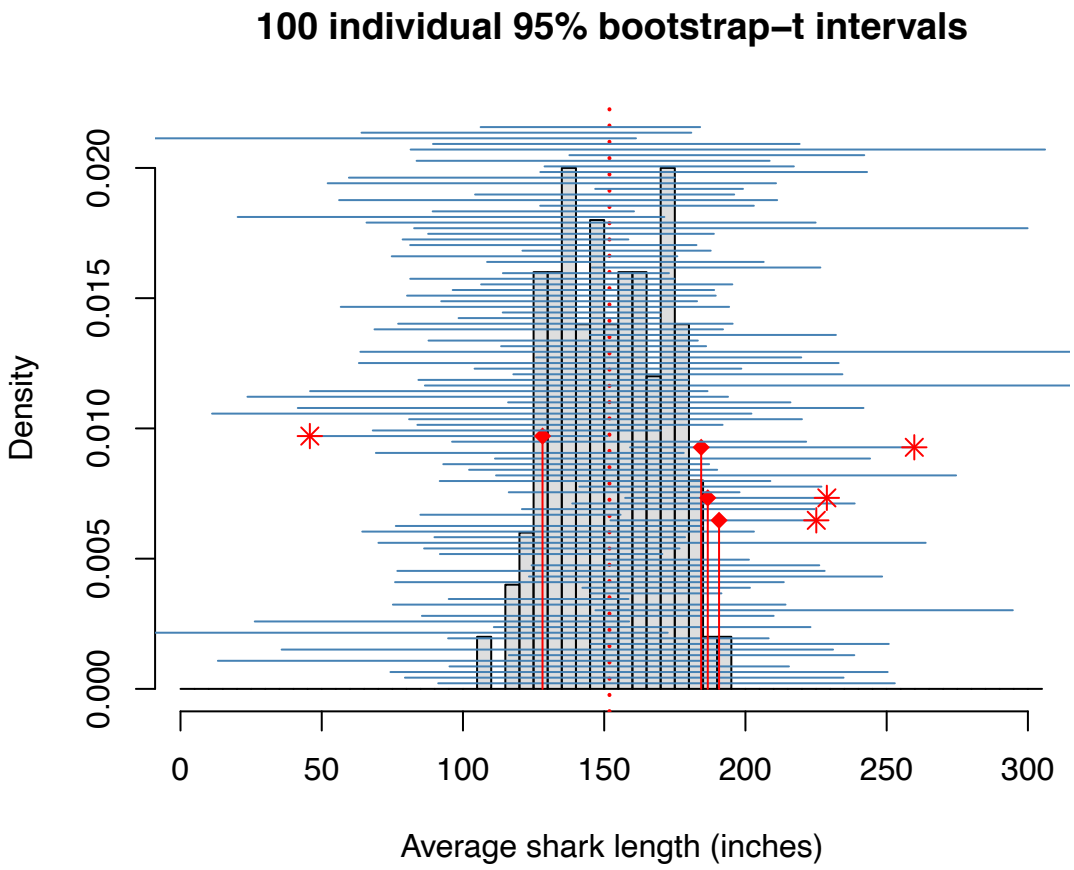


Figure 1: General bootstrap-t intervals

The Percentile Method

- In Section 4.3 (whether you realized it or not), the confidence interval endpoints always corresponded to relevant quantiles of a sampling distribution
  - e.g., the endpoints for a 95% confidence interval for a population average were based on the 2.5^{th} and 97.5^{th} quantiles of the $N(0,1)$ or t_{n-1} distributions
  - Recall, these distributions were assumed to approximate the sampling distribution of the sample mean

So why not simply use quantiles from the bootstrap distribution to directly construct a confidence interval?

- The **percentile method** for bootstrap confidence intervals is the following:
  - For a given sample $\mathcal{S}$ generate B bootstrap samples $\mathcal{S}_1^\star, \ldots, \mathcal{S}_B^\star$ by sampling with replacement from the sample $\mathcal{S}$.
  - For the b^{th} bootstrap sample ($b = 1, ..., B$), calculate $a_b = a(\mathcal{S}_b^\star)$.
  - From the values $a_1, \ldots, a_B$, find $a_{lower} = Q_a\,(p/2)$ and $a_{upper} = Q_a\,(1 - p/2)$
  - Then the $100(1-p)\%$ confidence interval is $[a_{lower}, a_{upper}]$.

This approach is **equivariant** to any 1:1 transformation of the attribute, say $T(a(\mathcal{P}))$.

  - For an increasing function $T(\cdot)$ the corresponding interval for $T(a(\mathcal{P}))$ is simply $[T(a_{lower}), T(a_{upper})]$.
  - For an decreasing function $T(\cdot)$ the corresponding interval for $T(a(\mathcal{P}))$ is simply $[T(a_{upper}), T(a_{lower})]$.
  - So, we only need to determine the values a_{lower} and a_{upper} once for $a(\mathcal{P})$ and we have them for any $T(a(\mathcal{P}))$.

Example: Average Shark Length

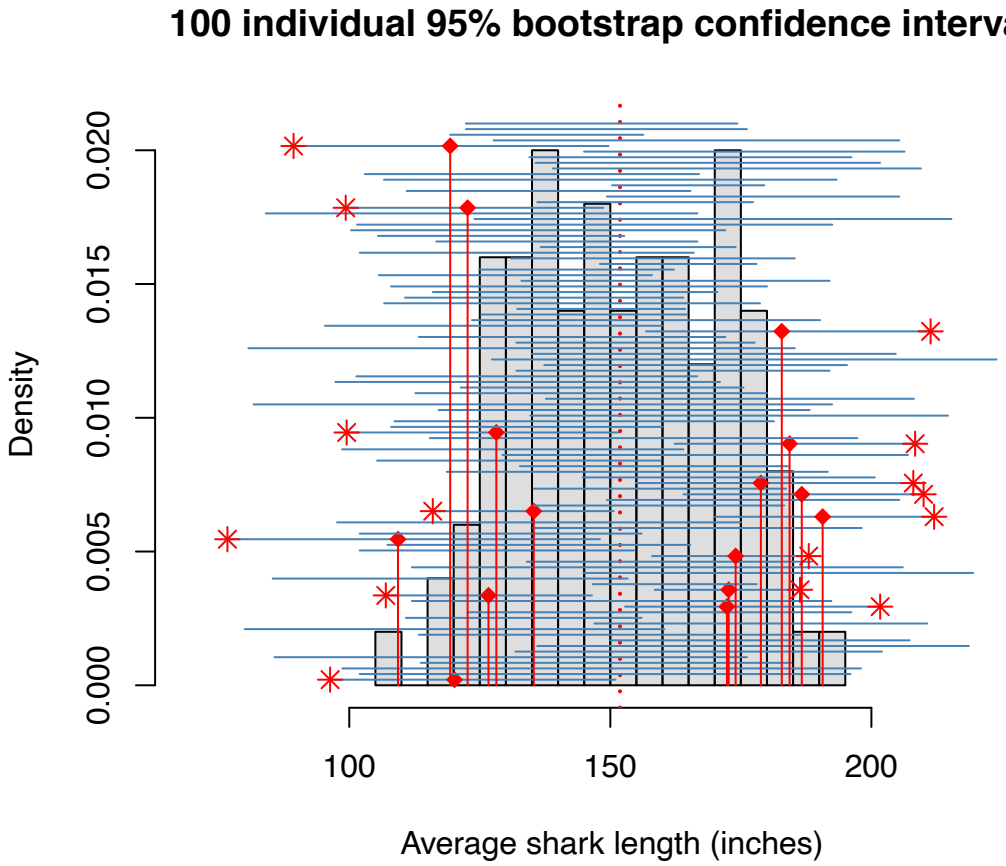
- Using the quantiles from the bootstrap distribution (based on 10,000 resamples from the sample) we obtain the following estimate of the interval that covers 95% of the bootstrap averages:
  - (94.83, 151.83)

- Note, the population average is 151.86

Estimating Coverage Probability: The Percentile Method

- To estimate the coverage probability associated with the percentile method (using a 95% confidence level) we
  - generate $B = 100$ samples of size $n = 6$ and for each sample calculate $a_b = a(\mathcal{S}_b^\star)$.
  - and then from the values $a_1, \ldots, a_B$, find $a_{lower} = Q_a\,(0.025)$ and $a_{upper} = Q_a\,(0.975)$
  - the 95% confidence interval for each sample is $[a_{lower}, a_{upper}]$.

- These 100 intervals (and their coverage) are illustrated below



- As can be seen, 85 of the 100 bootstrap intervals actually cover the average in the population.
  - This is a much lower coverage proportion than the expected 95% suggested

Comments:

- Simplicity is the attraction of this method, and explains its continued popularity.
- This method is transformation equivariant
- The coverage probability is often incorrect if the distribution of the estimator is not nearly symmetric
  - Coverage may be improved by considering "bias-corrected" versions of the percentile method. This, however, is outside the scope of this course. For more information on these alternatives see Section 11.3 in the book Computer Age Statistical Inference by Efron and Hastie.