10.3 Bootstrap

Bootstrap is used to estimate population parameters by Monte Carlo simulations when it is too difficult to do it analytically. When computers became powerful enough to handle large-scale simulations, the bootstrap methods got very popular for their ability to evaluate properties of various estimates.

Consider, for example, the *standard errors*. From the previous chapters, we know the standard errors of a sample mean and a sample proportion,

$$s(\overline{X}) = \frac{\sigma}{\sqrt{n}} \text{ and } s(\widehat{p}) = \sqrt{\frac{p(1-p)}{n}},$$

and they are quite simple. Well, try to derive standard errors of other statistics - sample median, sample variance, sample interquartile range, and so on. Each of these will require a substantial amount of work.

Many complicated estimates are being used nowadays in modern Statistics. How can one evaluate their performance, estimate their standard error, bias, etc.?

The difficulty is that we observe an estimator $\widehat{\theta}$ only once. That is, we have one sample $\mathcal{S} = (X_1, \ldots, X_n)$ from the population $\mathcal{P}$, and from this sample we compute $\widehat{\theta}$. We would very much like to observe many $\widehat{\theta}$'s and then compute their sample variance, for example, but we don't have this luxury. From one sample, we observe only one $\widehat{\theta}$, and this is just not enough to compute its sample variance!

10.3.1 Bootstrap distribution and all bootstrap samples

In 1970s, an American mathematician *Bradley Efron*, Professor at Stanford University, proposed a rather simple approach. He called it **bootstrap** referring to the idiom *"to pull oneself up by one's bootstraps"*, which means to find a solution relying on your own sources and without any help from outside (a classical example is Baron Münchausen from the 18th century collection of tales by R. E. Raspe, who avoided drowning in a lake by pulling himself from the water by his own hair!). In our situation, even though we really need several samples to explore the properties of an estimator $\widehat{\theta}$, we'll manage to do it with what we have, which is just one sample.

For the invention of bootstrap, Professor Efron is awarded the 2019 International Prize in Statistics which is said to be an equivalent of the Nobel Prize in Statistics.

To start, let's notice that many commonly used statistics are constructed in the same way as the corresponding population parameters. We are used to estimating a population mean μ by a sample mean $\overline{X}$, a population variance σ^2 by a sample variance s^2, population quantiles q_p by sample quantiles $\widehat{q}_p$, and so on. To estimate a certain parameter θ, we collect a random sample $\mathcal{S}$ from $\mathcal{P}$ and essentially compute the estimator $\widehat{\theta}$ from this sample by the same mechanism as θ was computed from the whole population!

In other words, there is a function g that one can use to compute a parameter θ from the population $\mathcal{P}$ (Figure 10.5). Then

$$\theta = g(\mathcal{P}) \text{ and } \widehat{\theta} = g(\mathcal{S}).$$

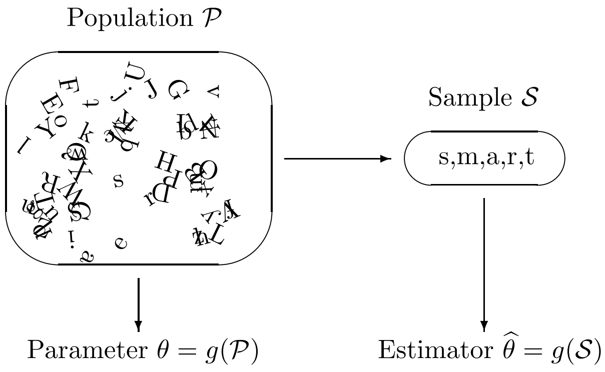
Population $\mathcal{P}$
Sample $\mathcal{S}$
s,m,a,r,t
Parameter $\theta = g(\mathcal{P})$
Estimator $\widehat{\theta} = g(\mathcal{S})$

FIGURE 10.5: *Parameter θ and its estimator $\widehat{\theta}$ computed by the same mechanism g applied to the population and to the sample.*

Example 10.17 (Population mean and sample mean). Imagine a strange calculator that can only do one operation g – averaging. Give it 4 and 6, and it returns their average $g(4,6) = 5$. Give it 3, 7, and 8, and it returns $g(3,7,8) = 6$.

Give it the whole population $\mathcal{P}$, and it returns the parameter $\theta = g(\mathcal{P}) = \mu$. Give it a sample $\mathcal{S} = (X_1, \ldots, X_n)$, and it returns the estimator $\widehat{\theta} = g(X_1, \ldots, X_n) = \overline{X}$. ◊

You can imagine similar calculators for the variances, medians, and other parameters and their estimates. Essentially, each estimator $\widehat{\theta}$ is a mechanism that copies the way a parameter θ is obtained from the population and then applies it to the sample.

Bradley Efron proposed one further step. Suppose some estimator is difficult to figure out. For example, we are interested in the variance of a sample median, $\eta = \operatorname{Var}(\widehat{M}) = h(\mathcal{P})$. This mechanism h consists of taking all possible samples from the population, taking their sample medians $\widehat{M}$, and then calculating their variance,

$$\eta = h(\mathcal{P}) = \mathbf{E}(\widehat{M} - \mathbf{E}\widehat{M})^2.$$

Of course, we typically cannot observe all possible samples; that's why we cannot compute $h(\mathcal{P})$, and parameter η is unknown. How can we estimate it based on just one sample $\mathcal{S}$?

Efron's **bootstrap approach** is to apply the same mechanism to the sample $\mathcal{S}$. That is, take all possible samples from $\mathcal{S}$, compute their medians, and then compute the sample variance of those, as on Figure 10.6. It may sound strange, but yes, we are proposing to create more samples from one sample $\mathcal{S}$ given to us. After all, exactly the same algorithm was used to compute $\eta = h(\mathcal{P})$. The advantage is that we know the observed sample $\mathcal{S}$, and therefore, we can perform all the bootstrap steps and estimate η.

Sampling from the observed sample $\mathcal{S}$ is a statistical technique called *resampling*. Bootstrap is one of the resampling methods. The obtained samples from $\mathcal{S}$ are called *bootstrap samples*. Each bootstrap sample $\mathcal{B}_j = (X^*_{1j}, \ldots, X^*_{nj})$ consists of values X^*_{ij} sampled from $\mathcal{S} = (X_1, \ldots, X_n)$ independently and with equal probabilities. That is,

$$X^*_{ij} = \begin{cases} X_1 & \text{with probability } 1/n \\ X_2 & \text{with probability } 1/n \\ \cdots & \cdots \quad \cdots \quad \cdots \quad \cdots \\ X_n & \text{with probability } 1/n \end{cases}$$

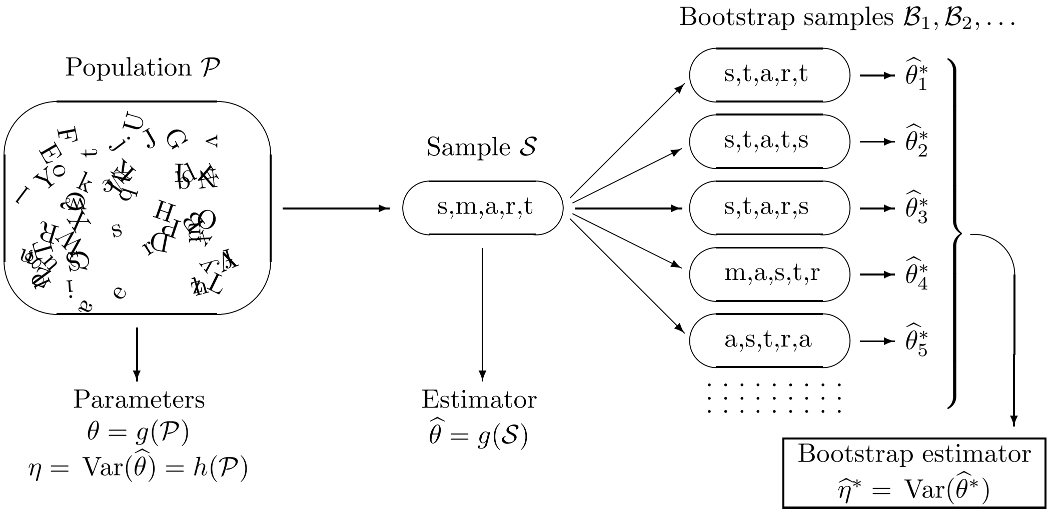

FIGURE 10.6: *Bootstrap procedure estimates* $\eta = \mathrm{Var}(\widehat{\theta})$ *by the variance of* $\widehat{\theta}_i^*$*'s, obtained from bootstrap samples.*

This is *sampling with replacement*, which means that the same observation X_i can be sampled more than once. An asterisk $(^*)$ is used to denote the contents of bootstrap samples.

DEFINITION 10.3

A **bootstrap sample** is a random sample drawn with replacement from the observed sample $\mathcal{S}$ of the same size as $\mathcal{S}$.

The distribution of a statistic across bootstrap samples is called a **bootstrap distribution**.

An estimator that is computed on basis of bootstrap samples is a **bootstrap estimator**.

Example 10.18 (Variance of a sample median). Suppose that we observed a small sample $\mathcal{S} = (2, 5, 7)$ and estimated the population median M with the sample median $\widehat{M} = 5$. How can we estimate its variance $\mathrm{Var}(\widehat{M})$?

Solution. Table 10.1 lists all $3^3 = 27$ equally likely bootstrap samples that can be drawn from $\mathcal{S}$. Among these, 7 samples have $\widehat{M}_i^* = 2$, 13 samples have $\widehat{M}_i^* = 5$, and 7 samples have $\widehat{M}_i^* = 7$. So, the *bootstrap distribution* of a sample median is

$$P^*(2) = 7/27, \quad P^*(5) = 13/27, \quad P^*(7) = 7/27. \qquad (10.8)$$

i	$\mathcal{B}_i$	$\widehat{M}_i$
1	$(2,2,2)$	2
2	$(2,2,5)$	2
3	$(2,2,7)$	2
4	$(2,5,2)$	2
5	$(2,5,5)$	5
6	$(2,5,7)$	5
7	$(2,7,2)$	2
8	$(2,7,5)$	5
9	$(2,7,7)$	7
10	$(5,2,2)$	2
11	$(5,2,5)$	5
12	$(5,2,7)$	5
13	$(5,5,2)$	5
14	$(5,5,5)$	5
15	$(5,5,7)$	5
16	$(5,7,2)$	5
17	$(5,7,5)$	5
18	$(5,7,7)$	7
19	$(7,2,2)$	2
20	$(7,2,5)$	5
21	$(7,2,7)$	7
22	$(7,5,2)$	5
23	$(7,5,5)$	5
24	$(7,5,7)$	7
25	$(7,7,2)$	7
26	$(7,7,5)$	7
27	$(7,7,7)$	7

TABLE 10.1: *All bootstrap samples $\mathcal{B}_i$ drawn from $\mathcal{S}$ and the corresponding sample medians for Example 10.18.*

We use it to estimate $h(\mathcal{P}) = \mathrm{Var}(\widehat{M})$ with the *bootstrap estimator*

$$\begin{aligned}
\widehat{\mathrm{Var}}^*(\widehat{M}) &= h(\mathcal{S}) = \sum_x x^2 P^*(x) - \left(\sum_x x P^*(x)\right)^2 \\
&= (4)\left(\frac{7}{27}\right) + (25)\left(\frac{13}{27}\right) + (49)\left(\frac{7}{27}\right) - \left\{(2)\left(\frac{7}{27}\right) + (5)\left(\frac{13}{27}\right) + (7)\left(\frac{7}{27}\right)\right\}^2 \\
&= \underline{3.303}.
\end{aligned}$$

◊

Here is a summary of the bootstrap method that we applied in this example.

Bootstrap (all bootstrap samples)

To estimate parameter η of the distribution of $\widehat{\theta}$:
1. Consider all possible bootstrap samples drawn with replacement from the given sample $\mathcal{S}$ as well as statistics $\widehat{\theta}^*$ computed from them.
2. Derive the bootstrap distribution of $\widehat{\theta}^*$.
3. Compute the parameter of this bootstrap distribution that has the same meaning as η.

All right, one may say, this certainly works for a "toy" sample of size three. But how about bigger samples? We can list all n^n possible bootstrap samples for very small n. However, a rather modest sample of size $n = 60$ can produce $60^{60} \approx 4.9 \cdot 10^{106}$ different bootstrap samples, which is almost five million times larger than *the googol*!

Certainly, we are not going to list a googol of bootstrap samples. Instead, we'll discuss two alternative approaches. The first one proposes to compute the bootstrap distribution without listing all the bootstrap samples. This, however, is still feasible only in relatively simple situations. The second solution, by far the most popular one, uses Monte Carlo simulations to produce a large number b of bootstrap samples. Although this way we may not get *all* possible bootstrap samples, results will be very close for large b.

Bootstrap distribution

The only reason for considering all possible bootstrap samples in Example 10.18 was to find the distribution (10.8) and then to obtain the bootstrap estimator $\widehat{\eta}^* = \widehat{\mathrm{Var}}(\widehat{M}^*)$ from it.

Sometimes it is possible to compute the bootstrap distribution without listing all bootstrap samples. Here is an example.

Example 10.19 (Bias of a sample median). A sample median may be biased or unbiased for estimating the population median. It depends on the underlying distribution of data. Suppose we observed a sample

$$\mathcal{S} = (3,\ 5,\ 8,\ 5,\ 5,\ 8,\ 5,\ 4,\ 2).$$

Find a bootstrap estimator of $\eta = \mathrm{Bias}(\widehat{M})$, the bias of the sample median.

<u>Solution</u>. First, find the bootstrap distribution of a sample median $\widehat{M}^*$. Based on the given sample of size $n = 9$, the sample median of bootstrap samples can be equal to 2, 3, 4, 5, or 8. Let us compute the probability of each value.

Sampling from $\mathcal{S}$, values $X^*_{ij} = 2$, 3, and 4 appear with probability 1/9 because only one of each of them appears in the given sample. Next, $X^*_{ij} = 5$ with probability 4/9, and $X^*_{ij} = 8$ with probability 2/9.

Now, the sample median $\widehat{M}^*_i$ in any bootstrap sample $\mathcal{B}_i$ is the central or the 5th smallest observation. Thus, it equals 2 if at least 5 of 9 values in $\mathcal{B}_i$ equal 2. The probability of that is

$$P^*(2) = \boldsymbol{P}(Y \geq 5) = \sum_{y=5}^{9} \binom{9}{y} \left(\frac{1}{9}\right)^y \left(\frac{8}{9}\right)^{9-y} = 0.0014$$

for a Binomial$(n = 9, p = 1/9)$ variable Y.

Similarly, $\widehat{M}^*_i \leq 3$ if at least 5 of 9 values in $\mathcal{B}_i$ do not exceed 3. The probability of that is

$$F^*(3) = \boldsymbol{P}(Y \geq 5) = \sum_{y=5}^{9} \binom{9}{y} \left(\frac{2}{9}\right)^y \left(\frac{7}{9}\right)^{9-y} = 0.0304$$

for a Binomial$(n = 9, p)$ variable Y, where $p = 2/9$ is a probability to sample either $X^*_{ij} = 2$ or $X^*_{ij} = 3$.

Proceeding in a similar fashion, we get

$$\begin{aligned} F^*(4) &= \sum_{y=5}^{9} \binom{9}{y} \left(\frac{3}{9}\right)^y \left(\frac{6}{9}\right)^{9-y} = 0.1448, \\ F^*(5) &= \sum_{y=5}^{9} \binom{9}{y} \left(\frac{7}{9}\right)^y \left(\frac{2}{9}\right)^{9-y} = 0.9696, \text{ and} \\ F^*(8) &= 1. \end{aligned}$$

From this cdf, we can find the bootstrap probability mass function of $\widehat{M}_i^*$,

$$P^*(2) = 0.0014, \quad P^*(3) = 0.0304 - 0.0014 = 0.0290, \quad P^*(4) = 0.1448 - 0.0304 = 0.1144,$$

$$P^*(5) = 0.9696 - 0.1448 = 0.8248, \quad P^*(8) = 1 - 0.9696 = 0.0304.$$

From this, the bootstrap estimator of $\mathbf{E}(\widehat{M})$ is the expected value of the bootstrap distribution,

$$\mathbf{E}^*(\widehat{M}_i^*) = (2)(0.0014) + (3)(0.0290) + (4)(0.1144) + (5)(0.8248) + (8)(0.0304) = 4.9146.$$

Last step! The bias of $\widehat{M}$ is defined as $h(\mathcal{P}) = \text{Bias}(\widehat{M}) = \mathbf{E}(\widehat{M}) - M$. We have estimated the first term, $\mathbf{E}(\widehat{M})$. Following the bootstrap ideas, what should we use as an estimator of M, the second term of the bias? The answer is simple. We have agreed to estimate $g(\mathcal{P})$ with $g(\mathcal{S})$, so we just estimate the population median $g(\mathcal{P}) = M$ with the sample median $g(\mathcal{S}) = \widehat{M} = 5$ obtained from our original sample $\mathcal{S}$.

The bootstrap estimator of $\text{Bias}(\widehat{M}) = \mathbf{E}(\widehat{M}) - M$ (based on all possible bootstrap samples) is

$$\eta(\mathcal{S}) = \widehat{\text{Bias}}(\widehat{M}) = \mathbf{E}^*(\widehat{M}) - \widehat{M} = 4.9146 - 5 = \boxed{-0.0852}\,.$$

♢

Although the sample in Example 10.19 is still rather small, the method presented can be extended to a sample of any size. Manual computations may be rather tedious here, but one can write a suitable computer code.

10.3.2 Computer generated bootstrap samples

Modern Statistics makes use of many complicated estimates. As the earliest examples, Efron used bootstrap to explore properties of the sample correlation coefficient, the trimmed mean[1], and the excess error[2], but certainly, there are more complex situations. Typically, samples will be too large to list all the bootstrap samples, as in Table 10.1, and the statistics will be too complicated to figure out their bootstrap distributions, as in Example 10.19.

This is where *Monte Carlo simulations* kick in. Instead of listing all possible bootstrap samples, we use a computer to generate a large number b of them. The rest follows our general scheme on Figure 10.6, p. 342.

Bootstrap (generated bootstrap samples)

To estimate parameter η of the distribution of $\widehat{\theta}$:

1. Generate a large number b of bootstrap samples drawn with replacement from the given sample $\mathcal{S}$.
2. From each bootstrap sample $\mathcal{B}_i$, compute statistic $\widehat{\theta}_i^*$ the same way as $\widehat{\theta}$ is computed from the original sample $\mathcal{S}$.
3. Estimate parameter η from the obtained values of $\widehat{\theta}_1^*, \ldots, \widehat{\theta}_b^*$.

[1] Trimmed mean is a version of a sample mean, where a certain portion of the smallest and the largest observations is dropped before computing the arithmetic average. Trimmed means are not sensitive to a few extreme observations, and they are used if extreme outliers may be suspected in the sample.

[2] This measures how well one can estimate the error of prediction in regression analysis. We study regression in the next chapter.

This is a classical bootstrap method of evaluating properties of parameter estimates. Do you think it is less accurate than the one based on *all* the bootstrap samples? Notice that b, the number of generated bootstrap samples, can be very large. Increasing b gives more work to your computer, but it does not require a more advanced computer code or a larger original sample. And of course, as $b \to \infty$, our estimator of η becomes just as good as if we had a complete list of bootstrap samples. Typically, thousands or tens of thousands of bootstrap samples are being generated.

Software notes

The following MATLAB code generates b bootstrap samples from the given sample $\boldsymbol{X} = (X_1, \ldots, X_n)$.

— MATLAB ————

```
n = length(X);              % Sample size
U = ceil(n*rand(b,n));      % A b×n matrix of random integers from 1 to n
B = X(U);                   % A matrix of bootstrap samples.
                            % The i-th bootstrap sample is in the i-th row.
```

For example, based on a sample $\boldsymbol{X} = (10, 20, 30, 40, 50)$ and generated matrix U of random indices, we obtain a matrix B of bootstrap samples,

$$U = \begin{pmatrix} 1 & 4 & 5 & 5 & 1 \\ 3 & 2 & 3 & 1 & 5 \\ 3 & 4 & 5 & 2 & 3 \\ 1 & 4 & 1 & 2 & 3 \\ 2 & 4 & 3 & 5 & 1 \\ 1 & 3 & 1 & 3 & 5 \\ 4 & 1 & 5 & 5 & 4 \\ 2 & 2 & 1 & 1 & 2 \\ 3 & 5 & 4 & 2 & 3 \\ 1 & 1 & 5 & 1 & 3 \end{pmatrix}, \qquad B = \begin{pmatrix} \mathcal{B}_1 \\ \mathcal{B}_2 \\ \mathcal{B}_3 \\ \mathcal{B}_4 \\ \mathcal{B}_5 \\ \mathcal{B}_6 \\ \mathcal{B}_7 \\ \mathcal{B}_8 \\ \mathcal{B}_9 \\ \mathcal{B}_{10} \end{pmatrix} = \begin{pmatrix} 10 & 40 & 50 & 50 & 10 \\ 30 & 20 & 30 & 10 & 50 \\ 30 & 40 & 50 & 20 & 30 \\ 10 & 40 & 10 & 20 & 30 \\ 20 & 40 & 30 & 50 & 10 \\ 10 & 30 & 10 & 30 & 50 \\ 40 & 10 & 50 & 50 & 40 \\ 20 & 20 & 10 & 10 & 20 \\ 30 & 50 & 40 & 20 & 30 \\ 10 & 10 & 50 & 10 & 30 \end{pmatrix}$$

If b and n are so large that storing the entire matrices U and b requires too many computer resources, we can generate bootstrap samples in a do-loop, one at a time, and keep the statistics $\widehat{\theta}_i$ only.

In fact, MATLAB has a special command `bootstrp` for generating bootstrap samples and computing estimates from them. For example, the code

```
M = bootstrp(50000,@median,S);
```

takes the given sample $\mathcal{S}$, generates $b = 50,000$ bootstrap samples from it, computes medians from each of them, and stores these median in a vector M. After that, we can compute various sample statistics of M, the mean, standard deviation, etc., and use them to evaluate the properties of a sample median.

In R, generating b bootstrap samples and saving their medians is rather simple, making use of the command `sample`.

— R —

```
for (k in 1:B){                    # Do-loop to produce B bootstrap samples
b <- sample( n, n, replace=1 )     # Random subsample with replacement
BootMedian[k] <- median(X[b]) }    # Compute medians of bootstrap samples
```

After this, `sd(BootMedian)` returns the bootstrap estimator of $\mathrm{Std}(\widehat{M})$, the standard deviation of the sample median; `quantile(BootMedian,0.10)` is the tenth percentile of the distribution of $\widehat{M}$, etc.

Also, R has a special package `boot` for bootstrap inference. To take advantage of it, we have to define our statistic of interest as a function of a sample.

— R —

```
median.fn <- function(X,subsample){ return(median(X[subsample])) }
# Now we invoke package "boot" and apply it to our new function median.fn
install.packages("boot"); library(boot);
boot( X, median.fn, R=10000 )
```

This will generate $b = 10,000$ bootstrap samples from sample X, compute their medians, and use them to calculate bootstrap estimates of the bias and standard error of the sample median. Other bootstrap statistics can be computed from the whole set of generated bootstrap medians $\widehat{\theta}_1^*, \ldots, \widehat{\theta}_b^*$. We obtain it as component `t` of the object resulting from the `boot` command. For example,

```
BootMed <- boot(X, median.fn, R=10000)$t
```

Would you expect the same results when you run this code again? You should not. We realize that this estimation algorithm includes random number generation, and so, our results are random, and they will differ from each other. On the other hand, the differences between our bootstrap estimates will be small when the number of bootstrap samples b is large.

Example 10.20 (Bias of a sample median (continued)). Based on the sample

$$\mathcal{S} = (3,\ 5,\ 8,\ 5,\ 5,\ 8,\ 5,\ 4,\ 2)$$

from Example 10.19, we estimated the population median θ with the sample median $\widehat{\theta} = 5$. Next, we investigate the properties of $\widehat{\theta}$. These computer codes will estimate the bias of $\widehat{\theta}$, the standard error, the first quartile, and the probability of $\widehat{\theta} > 4$.

— R —

```
  S  <-  c(3,5,8,5,5,8,5,4,2);
  b  <-  100000;
 median.fn <- function(X,subsample){
        return(median(X[subsample])) }
 install.packages("boot"); library(boot);
 BootMed <- boot(X,median.fn,R=10000)$t
  biasM <- mean(M)-median(S)
  sterrM <- sd(M)
  q25 <- quantile(BootMed,0.25)
  prob4 <- mean(M > 4)
```

— Matlab —

```
S  =  [3 5 8 5 5 8 5 4 2];
b  =  100000;
M  =  bootstrp(b,@median,S);
biasM = mean(M)-median(S);
sterrM = std(M);
q25 = quantile(BootMed,0.25);
prob4 = mean(M > 4);
```

Vector M contains bootstrap statistics $\widehat{\theta}_1^*, \ldots, \widehat{\theta}_b^*$. Based on $b = 100,000$ bootstrap samples, we obtained

$$\begin{array}{rcccl}
\widehat{\text{Bias}^*}(\widehat{\theta}) & = & \overline{M} - \widehat{\theta} & = & -0.0858, \\
s^*(\widehat{\theta}) & = & s(M) & = & 0.7062, \qquad \text{and} \\
\widehat{\boldsymbol{P}}^*(\widehat{\theta} > 4) & = & \dfrac{\#\left\{ i \, : \, \widehat{\theta}_i^* > 4 \right\}}{b} & = & 0.8558.
\end{array}$$

$\Diamond$

10.3.3 Bootstrap confidence intervals

In the previous chapters, we learned how to construct confidence intervals for the population mean, proportion, variance, and also, difference of means, difference of proportions, and ratio of variances. Normal, T, χ^2, and F distributions were used in these cases. These methods required either a Normal distribution of the observed data or sufficiently large samples.

There are many situations, however, where these conditions will not hold, or the distribution of the test statistic is too difficult to obtain. Then the problem becomes nonparametric, and we can use *bootstrap* to construct approximately $(1 - \alpha)100\%$ confidence intervals for the population parameters.

Two methods of bootstrap confidence intervals are rather popular.

Parametric method, based on the bootstrap estimation of the standard error

This method is used when we need a confidence interval for a parameter θ, and its estimator $\widehat{\theta}$ has approximately Normal distribution.

In this case, we compute $s^*(\widehat{\theta})$, the bootstrap estimator of the standard error $\sigma(\widehat{\theta})$, and use it to compute the approximately $(1 - \alpha)100\%$ confidence interval for θ,

Parametric bootstrap confidence interval

$$\boxed{\widehat{\theta} \pm z_{\alpha/2} s^*(\widehat{\theta})} \qquad (10.9)$$

It is similar to our usual formula for the confidence interval in the case of Normal distribution (9.3), and $z_{\alpha/2} = q_{1-\alpha/2}$ in it, as always, denotes the $(1 - \alpha/2)$-quantile from the Standard Normal distribution.

Example 10.21 (Confidence interval for a correlation coefficient). Example 8.20 on p. 237 and data set `Antivirus` contain the data on the number of times X the antivirus software was launched on 30 computers during 1 month and the number Y of detected worms,

X	30	30	30	30	30	30	30	30	30	30	30	15	15	15	10
Y	0	0	1	0	0	0	1	1	0	0	0	0	1	1	0

X	10	10	6	6	5	5	5	4	4	4	4	4	1	1	1
Y	0	2	0	4	1	2	0	2	1	0	1	0	6	3	1

Scatter plots on Figure 8.11 showed existence of a negative correlation between X and Y, which means that in general, the number of worms reduces if the antivirus software is used more often.

Next, the computer manager asks for a 90% confidence interval for the correlation coefficient ρ between X and Y.

The following MATLAB code can be used to solve this problem. This is a detailed step-by-step solution that our readers can translate line by line to other software languages.

— Matlab ————

```
alpha = 0.10;
X = [30 30 30 30 30 30 30 30 30 30 30 15 15 15 10 10 10 6 6 5 5 5 4 4 4 4 4 1 1 1];
Y = [0 0 1 0 0 0 1 1 0 0 0 0 1 1 0 0 2 0 4 1 2 0 2 1 0 1 0 6 3 1];
r = corrcoef(X,Y); % correlation coefficient from the given sample
r = r(2,1); % because corrcoef returns a matrix of correlation coefficients
b = 10000; n = length(X);
U = ceil(n*rand(b,n));
BootX = X(U); BootY = Y(U); % Values X and Y of generated bootstrap samples
BootR = zeros(b,1); % Initiate the vector of bootstrap corr. coefficients
for i=1:b;
  BR = corrcoef(BootX(i,:),BootY(i,:));
  BootR(i) = BR(2,1);
end;
s = std(BootR); % Bootstrap estimator of the standard error of r
CI = [ r + s*norminv(alpha/2,0,1), r + s*norminv(1-alpha/2,0,1) ];
disp(CI) % Bootstrap confidence interval
```

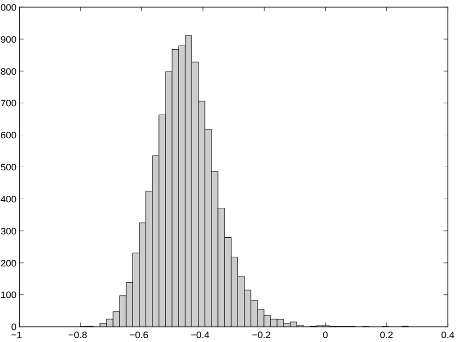

FIGURE 10.7: *The histogram of bootstrap correlation coefficients.*

As a result of this code, we get the sample correlation coefficient $r = -0.4533$, and also, $b = 10,000$ bootstrap correlation coefficients $r_1^*, \ldots, r_b^*$ obtained from b generated bootstrap samples.

Next, we notice that for the sample of size $n = 30$, r has approximately Normal distribution. For example, this can be confirmed by the histogram of bootstrap correlation coefficients on Figure 10.7.

Applying the parametric method, we compute $s^*(r) = 0.1028$, the standard error of $r_1^*, \ldots, r_b^*$, and use it to construct the 90% confidence interval

$$r \pm z_{\alpha/2} s^*(r) = -0.4533 \pm (1.645)(0.1028) = \underline{\text{[-0.6224, -0.2841]}}$$

This step-by-step MATLAB code can be translated to other software languages, including R. However, let's use the R package `boot` to see how we can handle the correlation coefficient which is a function of *two* variables X and Y instead of one.

We'll define a data frame `D` consisting of these two observed variables and then a function `correl` of it.

— R ————————

```
X <- c(rep(30,11),15,15,15,10,10,10,6,6,5,5,5,4,4,4,4,4,1,1,1)
Y <- c(0,0,1,0,0,0,1,1,0,0,0,0,1,1,0,0,2,0,4,1,2,0,2,1,0,1,0,6,3,1)
D <- data.frame(X,Y)
correl <- function(D,subsample){
X <- D[subsample,1]; Y <- D[subsample,2];
return( cor(X,Y) )
}
install.packages("boot"); library(boot);
BootR <- boot(data=D, statistic=correl, R=10000)$t
alpha <- 0.10; r <- cor(X,Y); s <- sd(BootR);
CI <- c( r + s*qnorm(alpha/2), r + s*qnorm(1-alpha/2) )
print(CI)
```

◊

Nonparametric method, based on the bootstrap quantiles

Equation 10.9 simply estimates two quantiles $q_{\alpha/2}$ and $q_{1-\alpha/2}$ of the distribution of statistic $\widehat{\theta}$, so that

$$\boldsymbol{P}\left\{q_{\alpha/2} \le \widehat{\theta} \le q_{1-\alpha/2}\right\} = 1 - \alpha. \qquad (10.10)$$

Since $\widehat{\theta}$ estimates parameter θ, this becomes an approximately $(1-\alpha)100\%$ confidence interval for θ.

This method fails if the distribution of $\widehat{\theta}$ is not Normal. The coverage probability in (10.10) may be rather different from $(1-\alpha)$ in this case. However, the idea to construct a confidence interval based on the quantiles of $\widehat{\theta}$ is still valid.

The quantiles $q_{\alpha/2}$ and $q_{1-\alpha/2}$ of the distribution of $\widehat{\theta}$ will then be estimated from the bootstrap samples. To do this, we generate b bootstrap samples, compute statistic $\widehat{\theta}^*$ for each of them, and determine the sample quantiles $\widehat{q}^*_{\alpha/2}$ and $\widehat{q}^*_{1-\alpha/2}$ from them. These quantiles become the end points of the $(1-\alpha)100\%$ confidence interval for θ.

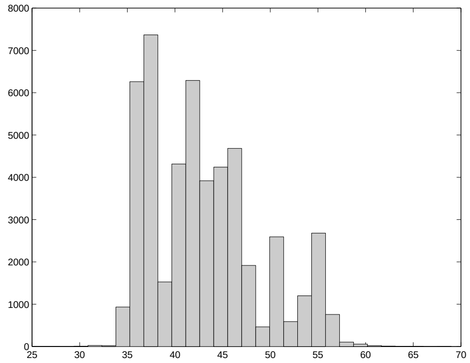

FIGURE 10.8: *The histogram of bootstrap medians.*

Nonparametric bootstrap confidence interval for parameter θ	$\left[\widehat{q}^*_{\alpha/2},\ \widehat{q}^*_{1-\alpha/2}\right]$, where $q^*_{\alpha/2}$ and $\widehat{q}^*_{1-\alpha/2}$ are quantiles of the distribution of $\widehat{\theta}$ estimated from bootstrap samples	(10.11)

Example 10.22 (Confidence interval for the median CPU time). The population median was estimated in Example 8.12 on p. 223, based on the following observed CPU times (data set `CPU`):

70	36	43	69	82	48	34	62	35	15
59	139	46	37	42	30	55	56	36	82
38	89	54	25	35	24	22	9	56	19

Let us now compute the 95% bootstrap confidence interval for the median CPU time.

— R —

```
alpha <- 0.05;
X <- c( 70, 36, 43, 69, 82, 48, 34, 62, 35, 15, 59, 139, 46, 37, 42,
30, 55, 56, 36, 82, 38, 89, 54, 25, 35, 24, 22, 9, 56, 19)
median.fn <- function(X,subsample){ return(median(X[subsample])) }
install.packages("boot"); library(boot);
BootMed <- boot(data=X, statistic=median.fn, R=50000)$t
print( c( quantile(BootMed,alpha/2), quantile(BootMed,1-alpha/2) ) )
```

— MATLAB —

```
alpha = 0.05;
X = [ 70 36 43 69 82 48 34 62 35 15 59 139 46 37 42 ...
      30 55 56 36 82 38 89 54 25 35 24 22 9 56 19]';
b = 50000; n = length(X);
U = ceil(n*rand(b,n)); BootX = X(U); BootM = zeros(b,1);
for i=1:b; BootM(i) = median(BootX(i,:)); end;
CI = [ quantile(BootM,alpha/2), quantile(BootM,1-alpha/2) ]
```

These programs generate $b = 50,000$ bootstrap samples and compute their sample medians $\widehat{\theta}_1^*, \ldots, \widehat{\theta}_b^*$ (variable `BootM`). Based on these sample medians, the 0.025- and 0.975-quantiles are calculated. The 95% confidence interval `CI` then stretches between these quantiles, from $\widehat{q}^*_{0.025}$ to $\widehat{q}^*_{0.975}$.

This algorithm results in a confidence interval [35.5, 55.5].

By the way, the histogram of bootstrap medians $\widehat{\theta}_1^*, \ldots, \widehat{\theta}_b^*$ on Figure 10.8 shows a rather non-Normal distribution. We were essentially forced to use the nonparametric approach.

◊

MATLAB has a special command `bootci` for the construction of bootstrap confidence intervals. The problem in Example 10.22 can be solved by just one command

```
bootci(50000,{@median,X},'alpha',0.05,'type','percentile')
```

where 0.05 denotes the α-level, and `'percentile'` requests a confidence interval computed by the method of percentiles. This is precisely the nonparametric method that we have just discussed. Replace it with type `'normal'` to obtain a parametric bootstrap confidence interval that is based on the Normal distribution (10.9).