Bootstrap Methods

(An introduction)

N. Torelli, L. Egidi, G. Di Credico

Fall 2020

University of Trieste

Resampling methods

The nonparametric bootstrap

The parametric bootstrap

Bootstrap-based confidence intervals

Resampling methods

The idea of resampling methods

Resampling methods are **computer-intensive methods** that employ simulation to carry out inferential conclusions for the data available.

In some sense, they replace mathematical formulas with computer simulation, though proving their validity requires quite sophisticated mathematics.

There are several such methods, but by far the most important are **bootstrap methods**. They are relatively modern, but their initial development predates the modern computer age!

(*Note: this lecture follows in particular the CASI book*)

The jackknife: introduction

The jackknife is, so to speak, the ancestor of the bootstrap. Its main usage is to obtain a nonparametric estimate of the standard error of an estimate, resulting in a simpler alternative to the delta method for complex functions of model parameters.

Let us consider a random sample $y_i, \ldots, y_n$, with $Y_i \sim F$, for some distribution F.

We are interested in a real-valued statistic (parameter estimate) $\widehat{\psi} = s(\mathbf{y})$, where $s(\cdot)$ is a given function of the n observations.

The jackknife: details

Let $\mathbf{y}_{(i)}$ be the sample without the i-th observation y_i

$$\mathbf{y}_{(i)} = (y_1, y_2, \ldots, y_{i-1}, y_{i+1}, \ldots, y_n)$$

so that $\widehat{\psi}_{(i)} = s(\mathbf{y}_{(i)})$ is the corresponding statistic of interest.

The jackknife estimate of standard error for $\widehat{\psi}$ is

$$\widehat{\mathrm{SE}}_{jack} = \left[\frac{n-1}{n} \sum_{i=1}^{n} \left(\widehat{\psi}_{(i)} - \widehat{\psi}_{(\cdot)} \right)^2 \right]^{1/2}, \quad \text{with} \quad \widehat{\psi}_{(\cdot)} = \frac{1}{n} \sum_{i=1}^{n} \widehat{\psi}_{(i)} \, .$$

Note: the formula works also when each observation y_i is multidimensional (e.g. for regression models).

The jackknife: comments

1. When $\widehat{\psi} = \overline{y}$, with some algebra we obtain that $\widehat{\mathrm{SE}}_{jack} = s/\sqrt{n}$, the usual estimated standard error of the mean. (This is the reason to introduce the factor $(n-1)/n$ in the definition).
2. The jackknife standard error is upwardly biased as an estimate of the true standard error. The bias disappears with larger n.
3. The important property of the procedure is that the definition can be applied to *any* statistic of interest, even very complex ones.

We only need an algorithm to compute $s(\mathbf{y})$: **computer power replaces the theoretical Taylor series calculations of the delta method**.

An example

Let us consider the standard error of the correlation coefficient for a random sample of bivariate normal data $(x_1, y_1)^\top, \ldots, (x_n, y_n)^\top$:

$$\widehat{\psi} = \frac{\sum_i (x_i - \overline{x})\,(y_i - \overline{y})}{\sqrt{\sum_i (x_i - \overline{x})^2}\,\sqrt{\sum_i (y_i - \overline{y})^2}}$$

This is an estimate of the corresponding parameter $\mathrm{cov}(X_i, Y_i)/(\sigma_x \sigma_y)$, and we can compute its standard error by the multidimensional version of the delta method.

The related formula is not exactly friendly (from CASI book, page 157)

$$\widehat{\mathrm{se}}_{\mathrm{taylor}} = \left\{ \frac{\hat{\theta}^2}{4n} \left[\frac{\hat{\mu}_{40}}{\hat{\mu}_{20}^2} + \frac{\hat{\mu}_{04}}{\hat{\mu}_{02}^2} + \frac{2\hat{\mu}_{22}}{\hat{\mu}_{20}\hat{\mu}_{02}} + \frac{4\hat{\mu}_{22}}{\hat{\mu}_{11}^2} - \frac{4\hat{\mu}_{31}}{\hat{\mu}_{11}\hat{\mu}_{20}} - \frac{4\hat{\mu}_{13}}{\hat{\mu}_{11}\hat{\mu}_{02}} \right] \right\}^{1/2} \tag{10.10}$$

where

$$\hat{\mu}_{hk} = \sum_{i=1}^{n} (x_i - \bar{x})^h (y_i - \bar{y})^k / n. \tag{10.11}$$

R lab: the jackknife at work I

As a first example, let us consider the case of the sample mean.

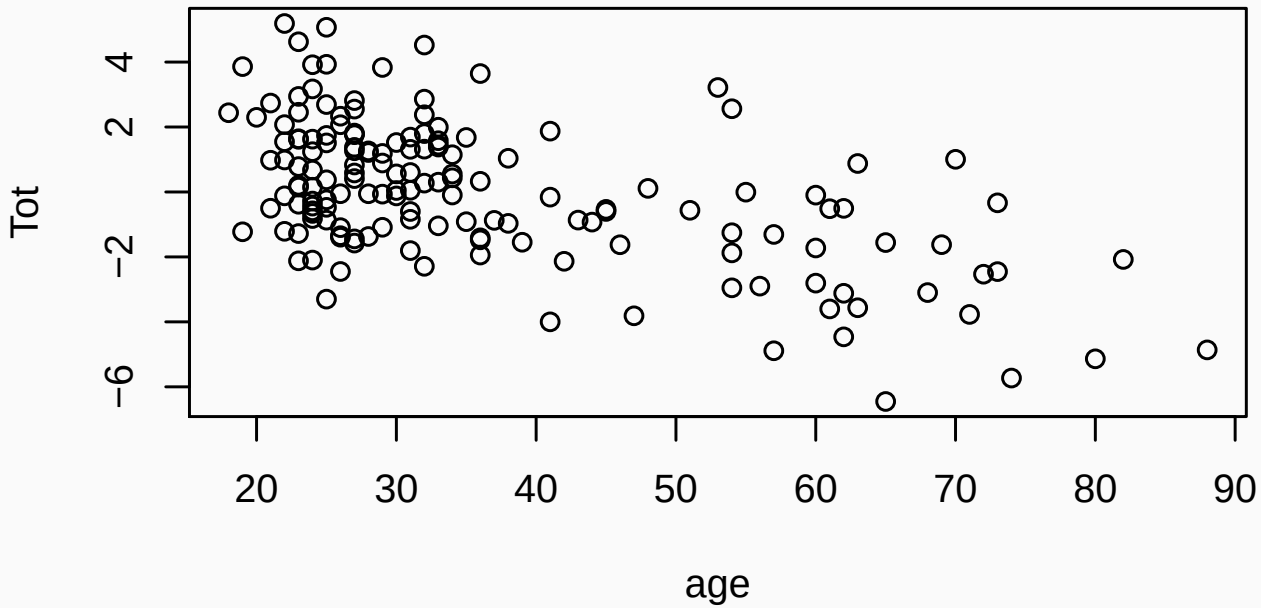
```
library(DAAG); y <- with(pair65, heated - ambient)
n <- length(y); s_vect <- rep(0, n)
for(i in 1:n) s_vect[i] <- mean(y[-i])
SE_jack <- sqrt(((n - 1)/n) * sum((s_vect - mean(s_vect))^2))
print(c(sd(y) / sqrt(n), SE_jack))
```

```
## [1] 2.034426 2.034426
```

R lab: the jackknife at work II

The second example concerns the correlation coefficient, and we use the same data set of the CASI book, the `kidneydata` dataset; here `Tot` is a composite measure of kidney overall function.

```
load("kidneydata.RData")
with(as.data.frame(kidneydata), plot(Tot ~ age))
```



R lab: the jackknife at work II

The standard error based on the delta method is stored in the variable `se_delta`, and it is taken from the CASI book. We note that `SE_jack` is slightly larger.

```
SE_delta <- 0.057
n <- nrow(kidneydata)
s_vect <- rep(0, n)
for(i in 1:n) s_vect[i] <- cor(kidneydata[-i, ])[1, 2]
SE_jack <- sqrt(((n - 1)/n) * sum((s_vect - mean(s_vect))^2))

print(c(SE_delta, SE_jack))
```

```
## [1] 0.05700000 0.05820618
```

The nonparametric bootstrap

Introduction

As reported in the CASI book, the jackknife lies *between classical methodology and a full-throated use of electronic computation*, whereas the bootstrap is an undisputed *computer-intensive* statistical method.

Another important difference is that the bootstrap has a rather wide scope of application, while instead the jackknife is mainly used for standard errors.

A legendary beginning

The two methods are indeed related, as testified by the paper that introduced the bootstrap.

The Annals of Statistics
1979, Vol. 7, No. 1, 1–26

THE 1977 RIETZ LECTURE

BOOTSTRAP METHODS: ANOTHER LOOK AT THE JACKKNIFE

BY B. EFRON

Stanford University

We discuss the following problem: given a random sample $\mathbf{X} = (X_1, X_2, \ldots, X_n)$ from an unknown probability distribution F, estimate the sampling distribution of some prespecified random variable $R(\mathbf{X}, F)$, on the basis of the observed data $\mathbf{x}$. (Standard jackknife theory gives an approximate mean and variance in the case $R(\mathbf{X}, F) = \theta(\hat{F}) - \theta(F)$, θ some parameter of interest.) A general method, called the "bootstrap," is introduced, and shown to work satisfactorily on a variety of estimation problems. The jackknife is shown to be a linear approximation method for the bootstrap. The exposition proceeds by a series of examples: variance of the sample median, error rates in a linear discriminant analysis, ratio estimation, estimating regression parameters, etc.

The bootstrap idea

The bootstrap idea is very simple, and to illustrate it we start from the same problem introduced for the jackknife, namely the estimation of the standard error of $\widehat{\psi} = s(\mathbf{y})$.

The standard error requires the computation of $\mathrm{var}(\widehat{\psi})$, something computable by drawing a large number of independent random samples from the true model F.

This is impossible, since F is unknown, so the bootstrap uses instead an estimate $\widehat{F}$ in place of F, and then it proceeds with the simulation.

In particular, when $\widehat{F}$ is the empirical distribution function (we met it in the very first class) a single simulated sample is obtained by **random selection with replacement** from the observed sample.

An example (from Boos and Stefanski, 2010, *Significance*)

Table 1. Random sample of 25 yearly incomes in thousands of dollars (ordered from lowest to highest)

1	4	6	12	13	14	18	19	20	22	23	24	26
31	34	37	46	47	56	61	63	65	70	97	385	

Figure 2: $n = 25$ adult male yearly incomes in a fictitious county

The data were actually generated from a known distribution, namely

$$Y_i \sim 30 \exp(Z_i), \qquad Z_i \sim N(0,1) \qquad i = 1, \ldots, 25$$

so that in this case we know the true distribution of the data (the population).

Example: two bootstrap samples

Nonparametric bootstrap treats the data of the previous table as the population and draws samples of size $n = 25$ (with replacement) from it.

```
y <- c(1, 4, 6, 12, 13, 14, 18, 19, 20, 22, 23, 24, 26, 31, 34,
       37, 46, 47, 56, 61, 63, 65, 70, 97, 385)
n <- length(y); set.seed(1989); B <- 10^4
boot.sample <- matrix(NA, nrow = B, ncol = n)
boot.sample[1,] <- sample(y, n, replace = TRUE)
boot.sample[2,] <- sample(y, n, replace = TRUE)
kable(boot.sample[1:2, 1:15])
```

22	20	4	34	70	13	24	70	13	63	18	12	46	6	23
65	31	24	4	34	65	37	19	34	4	70	70	1	97	97

The bootstrap at work

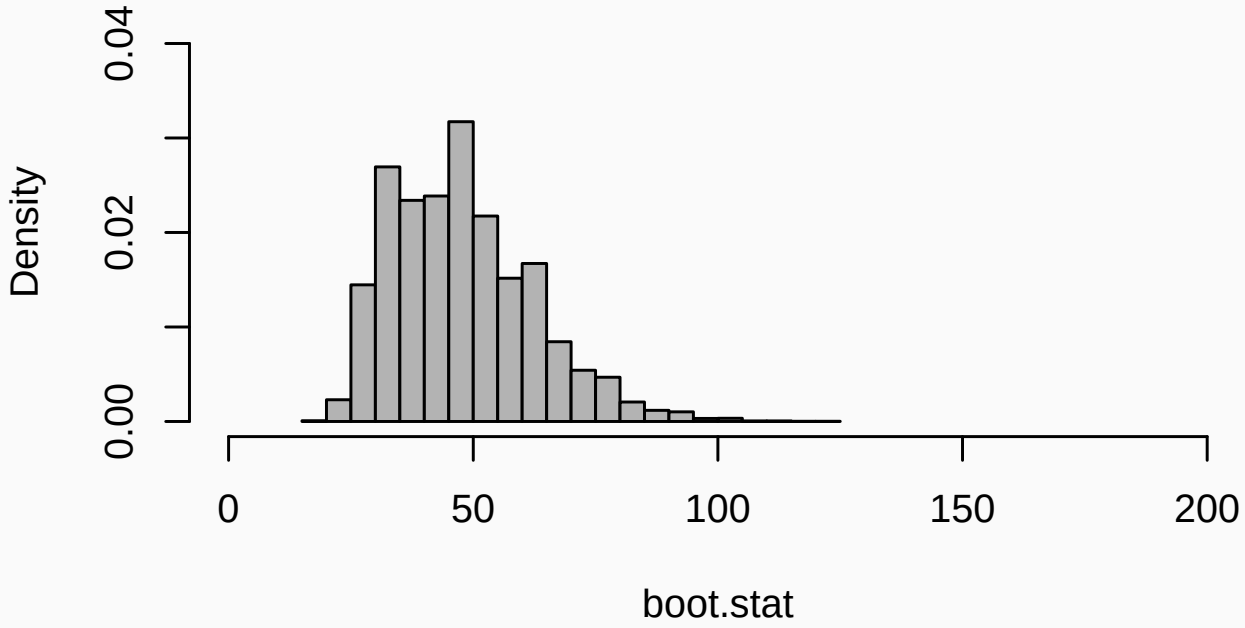
The bootstrap samples can be used to obtain an estimate of the standard error: denoted by $\widehat{\psi}^{*b}$, $b = 1, \ldots, B$ the statistic of interest for each bootstrap sample, we get

$$\widehat{\mathrm{SE}}_{boot} = \left[\frac{1}{B-1} \sum_{b=1}^{B} \left(\widehat{\psi}^{*b} - \widehat{\psi}^{*\cdot} \right)^2 \right]^{1/2}, \quad \text{with} \quad \widehat{\psi}^{*\cdot} = \frac{1}{B} \sum_{b=1}^{B} \widehat{\psi}^{*b}.$$

We can surely go beyond the computation of standard errors, since the set of bootstrap estimates $\widehat{\psi}^{*1}, \ldots, \widehat{\psi}^{*B}$ can be used to **approximate** the distribution of $\widehat{\psi}$.

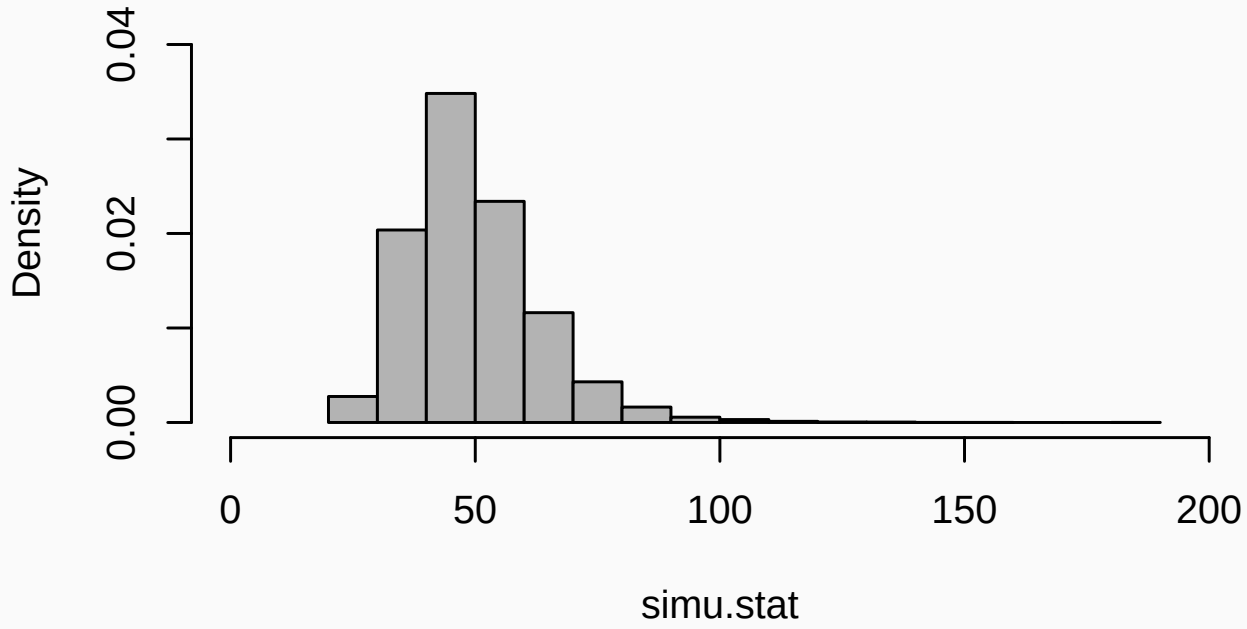
R lab: bootstrap distribution of $\widehat{\psi} = \overline{Y}$

```
B <- 10^4; boot.sample <- matrix(NA, nrow = B, ncol = n)
for(i in 1:B) boot.sample[i,] <- sample(y, n, replace = TRUE)
boot.stat <- rowMeans(boot.sample)
hist(boot.stat, main="", breaks=20, prob=TRUE, col=gray(0.7),
    xlim=c(0, 200), ylim=c(0, 0.04))
```



R lab: comparison with the true distribution

```
B <- 10^4; simu.sample <- matrix(NA, nrow = B, ncol = n)
for(i in 1:B) simu.sample[i,] <- mean(30 * exp(rnorm(n)))
simu.stat <- rowMeans(simu.sample)
hist(simu.stat, main="", breaks=20, prob=TRUE, col=gray(0.7),
     xlim=c(0, 200), ylim=c(0, 0.04))
```



Back to the standard error computation

Provide B is large enough, the bootstrap-based standard error is unbiased, thus outperforming the jackknife.

```
n <- nrow(kidneydata); B <- 10^4
s_vect <- rep(0, B)
for(i in 1:B) {ind <- sample(1:n, n, replace = TRUE)
               s_vect[i] <- cor(kidneydata[ind,])[1, 2]}
SE_boot <- sd(s_vect)

print(c(SE_delta, SE_jack, SE_boot))
```

```
## [1] 0.05700000 0.05820618 0.05820597
```

More on the boostrap idea

The bootstrap idea can be appreciated by noticing the parallel interpretation existing for the statistical model for the sample data

$$F \xrightarrow{\text{i.i.d.}} \mathbf{y} \xrightarrow{s(\cdot)} \widehat{\psi}$$

and the bootstrap mechanism

$$\widehat{F} \xrightarrow{\text{i.i.d.}} \mathbf{y}^* \xrightarrow{s(\cdot)} \widehat{\psi}^* \, .$$

The link between the two representations is given by the fact that $\widehat{F}$ **approaches the true** F **when** $n \to \infty$, which is the key fact.

Comments on nonparametric bootstrap

1. It is completely automatic! The underlying math is not simple, but it has been rigorously carried out.
2. It is large-sample method, since its accuracy increases with n.
3. Can be extended to any statistic of interest, not just estimated standard errors.
4. Can be extended to more complex settings, including some models with dependent data.
5. It also has some limitations, like being not appropriate for sample extremes, such as the minimum or maximum value of the observed data. (For these latter problems, there are some specific adjustments, but they are not simple).

The parametric bootstrap

Parametric bootstrap

Going back to the bootstrap mechanism, there is actually no need $\widehat{F}$ be the nonparametric estimate of F (the empirical cdf).

Another alternative is to assume a parametric statistical model $f_{\boldsymbol{\theta}}(\mathbf{y})$ for the data, and simulate the bootstrap samples from $f_{\hat{\boldsymbol{\theta}}}$, where as usual $\widehat{\boldsymbol{\theta}}$ is a point estimate of $\boldsymbol{\theta}$.

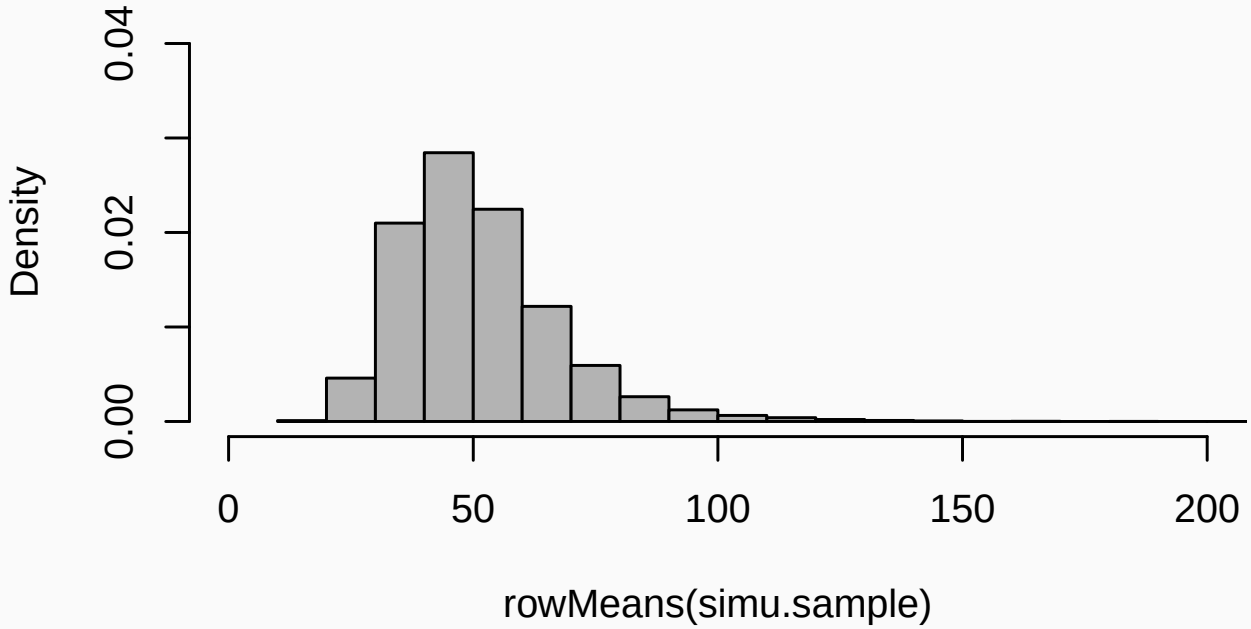
The mechanism becomes

$$f_{\hat{\boldsymbol{\theta}}} \longrightarrow \mathbf{y}^* \xrightarrow{s(\cdot)} \widehat{\psi}^*$$

Extensions to (very) complex models become easier, but a realistic model is required.

R lab: parametric bootstrap for $\widehat{\psi} = \overline{Y}$

```
n <- length(y); mu <- mean(log(y)); sigma <- sd(log(y))
simu.sample <- matrix(NA, nrow = B, ncol = n)
for(i in 1:B) simu.sample[i,] <- mean(exp(rnorm(n, mu, sigma)))
hist(rowMeans(simu.sample) , main="", breaks=25, prob=TRUE,
     col=gray(0.7), xlim=c(0, 200), ylim=c(0, 0.04))
```



Application to hypothesis testing

Parametric bootstrap can be employed for obtaining p-values by simulation, also for those cases when the model has some parameters that have to be estimated also under H_0.

For example, we can obtain a fairly good approximation to the exact p-value for the classic one sample t-test.

We only need to keep in mind that the bootstrap samples must be generated from the model estimated under H_0. This means that for testing

$$\begin{cases} H_0 : \mu = \mu_0 \\ H_1 : \mu \neq \mu_0 \end{cases}$$

for the usual i.i.d. normal model, we need to generate data with $\mu = \mu_0$ and $\sigma^2 = \widehat{\sigma}_0^2 = \sum_i (y_i - \mu_0)^2 / n$.

R lab: t-test by parametric bootstrap

```
library(DAAG); y <- with(pair65, heated - ambient)
n <- length(y)
z_obs <- mean(y) / sqrt(var(y) / n)
s0 <- sqrt(mean((y - 0)^2))
B <- 10000; z_sim <- numeric(B)
for(i in 1:B) { ys <- rnorm(n, m = 0, s = s0)
                z_sim[i] <- mean(ys) / sqrt(var(ys) / n)}
c(t.test(y)$p.val, mean(abs(z_sim) >= abs(z_obs)))
```

```
## [1] 0.01437832 0.01310000
```

Bootstrap-based confidence intervals

The bootstrap automation of confidence intervals

We mentioned the standard approximate Wald-type 95% confidence interval for a parameter of interest ψ, in a model with parameter $\boldsymbol{\theta}$:

$$\widehat{\psi} \pm 1.96\, \mathrm{SE}(\widehat{\psi})$$

This is widely used, but it has two shortcomings:

1. It requires the estimated standard error $\mathrm{SE}(\widehat{\psi})$, which may be hard to compute.
2. It is symmetric around the point estimate, and sometimes this leads to inaccuracy, since the finite sample distribution of $\widehat{\psi}$ (and hence of the related pivot) is often asymmetric.

The first point is solved by $\widehat{\mathrm{SE}}_{boot}$, but the bootstrap provides some further, more satisfactory solutions for confidence intervals.

Bootstrap-based confidence intervals

There are several available methods, and an extensive literature. Here we focus on the main ones (the approach is inspired by the MASS book), which are

1. The **percentile** method.
2. The **basic** method.
3. The **studentized** method.

These three methods work both for nonparametric and parametric bootstrap.

Further methods exist, such as the BC_a method, but they are used less often in practice.

Running example for confidence intervals

We use the `student score` dataset of the CASI book as a running example. It concerns the score of 22 students in 5 tests:

```
score <- read.table("figs/student_score.txt", header = TRUE)
print(cor(score))
```

```
##              mech      vecs       alg     analy      stat
## mech  1.0000000 0.4978075 0.7560364 0.6534763 0.5357744
## vecs  0.4978075 1.0000000 0.5922624 0.5071353 0.3786038
## alg   0.7560364 0.5922624 1.0000000 0.7627546 0.6698255
## analy 0.6534763 0.5071353 0.7627546 1.0000000 0.7376712
## stat  0.5357744 0.3786038 0.6698255 0.7376712 1.0000000
```

The parameter of interest is the *eigenratio* statistic for the above correlation matrix, namely $\psi =$ largest eigenvalue/ sum eigenvalues.

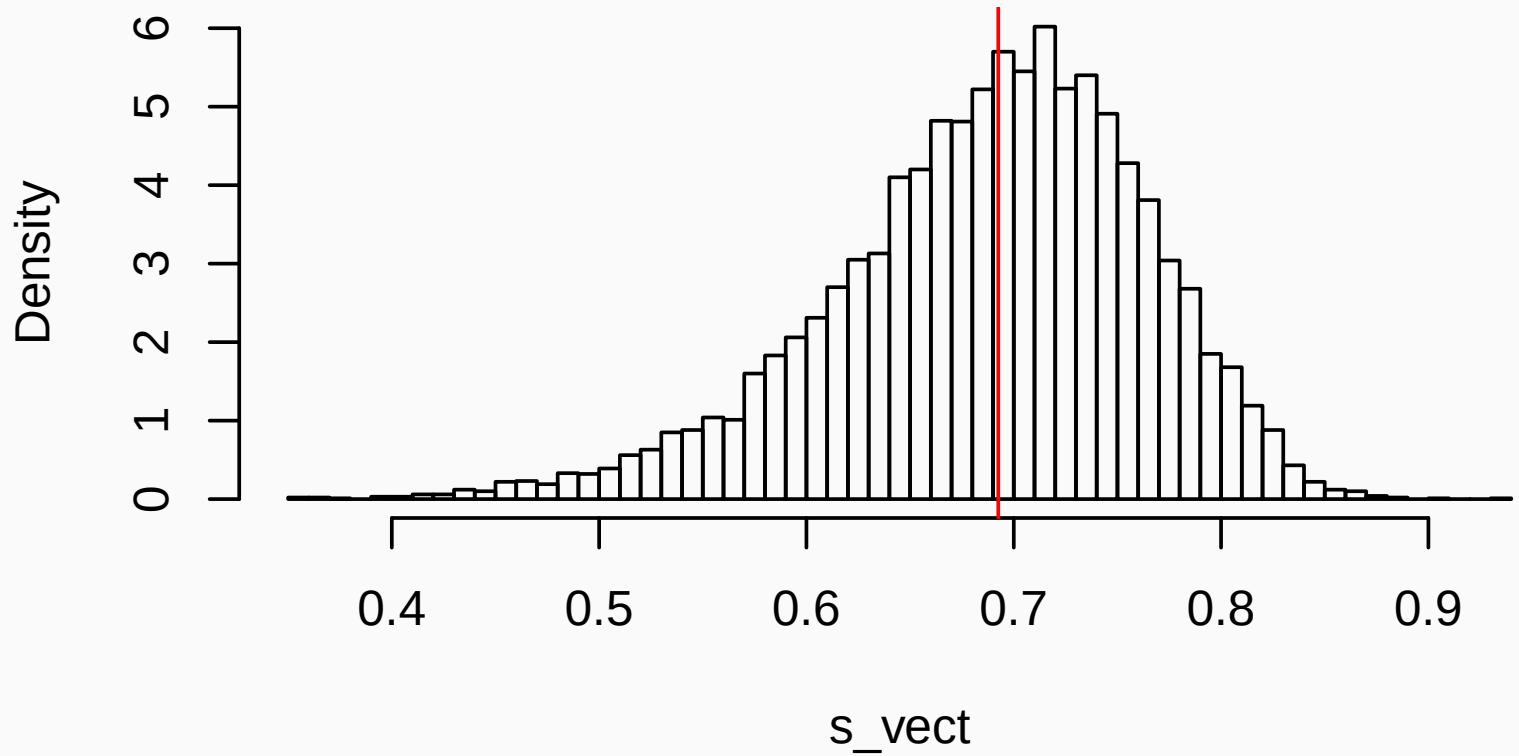
R lab: bootstrap (nonparametric) standard error for the student score data

```
psi_fun <- function(data) {eig <- eigen(cor(data))$values
                           return(max(eig) / sum(eig))}
psi_obs <-   psi_fun(score)
n <- nrow(score); B <- 10^4
s_vect <- rep(0, B)
for(i in 1:B) {ind <- sample(1:n, n, replace = TRUE)
               s_vect[i] <-  psi_fun(score[ind,])}
SE_boot <- sd(s_vect)
psi_obs + c(-1, 1) * 1.96 * SE_boot
```

```
## [1] 0.5448847 0.8401859
```

R lab: bootstrap distribution

```
hist.scott(s_vect, main = "")
abline(v = psi_obs, col = 2)
```



The percentile method

It simply uses the quantiles of the bootstrap distribution $\widehat{\psi}^{*1}, \ldots, \widehat{\psi}^{*B}$.

In the example, we get

```
perc_ci <- quantile(s_vect, prob=c(0.025, 0.975))
attr(perc_ci, "names") <- NULL
perc_ci
```

```
## [1] 0.5175641 0.8136610
```

Compared to the above Wald-type interval, and taking the point estimate as reference, the percentile confidence interval is wider on the left side and shorter on the right side.

The basic method

The basic intervals are based on the idea that the distribution of $\widehat{\psi}^* - \widehat{\psi}$ mimics that of $\widehat{\psi} - \psi$. If this is the case, we would get

$$0.95 = \Pr(L \leq \widehat{\psi} - \psi \leq U) \approx \Pr(L \leq \widehat{\psi}^* - \widehat{\psi} \leq U)$$

Using the first probability we obtain that a confidence interval for ψ is $(\widehat{\psi} - U, \widehat{\psi} - L)$, and then we use the second probability to obtain that $L + \widehat{\psi}$ and $U + \widehat{\psi}$ are estimated by the 2.5% and 97.5% bootstrap quantiles, respectively (here denoted by $q^*_{0.025}$ and $q^*_{0.975}$).

Putting the two things together we get the **basic bootstrap confidence interval**

$$(\widehat{\psi} - U, \widehat{\psi} - L) = (2\,\widehat{\psi} - q^*_{0.975}, 2\,\widehat{\psi} - q^*_{0.025})$$

R lab: basic confidence interval

```
basic_ci <- 2 * psi_obs - quantile(s_vect, prob=c(0.975, 0.025))
attr(basic_ci, "names") <- NULL
basic_ci
```

```
## [1] 0.5714096 0.8675066
```

Since the result is essentially the percentile interval reflected about $\widehat{\psi}$, the basic confidence interval is shorter on the left side and wider on the right side.

For asymmetric distributions of $\widehat{\psi}$ the basic confidence interval may have coverage probability closer to the target value than the percentile one. On the other hand, the percentile interval is invariant to monotonic transformations of ψ, and this is perhaps more important.

The studentized method

The last method is perhaps the most reliable of all the methods, but it requires a standard error estimate $\mathrm{SE}(\widehat{\psi}^*)$ from each bootstrap sample.

Denoting by $z^*_{0.025}$ and $z^*_{0.975}$ the bootstrap quantiles of $z^{*1}, \ldots, z^{*B}$, where $z^{*b} = (\widehat{\psi}^{*b} - \widehat{\psi})/\mathrm{SE}(\widehat{\psi}^{*b})$, the **studentized bootstrap confidence interval** is given by

$$(\widehat{\psi} - \mathrm{SE}(\widehat{\psi})\, z^*_{0.975}, \widehat{\psi} - \mathrm{SE}(\widehat{\psi})\, z^*_{0.025})$$

This is perhaps too challenging for the running example, since explicit estimates of $\mathrm{SE}(\widehat{\psi}^*)$ would be very hard. We could employ the jackknife within each bootstrap sample, or a *double bootstrap* scheme, though ...