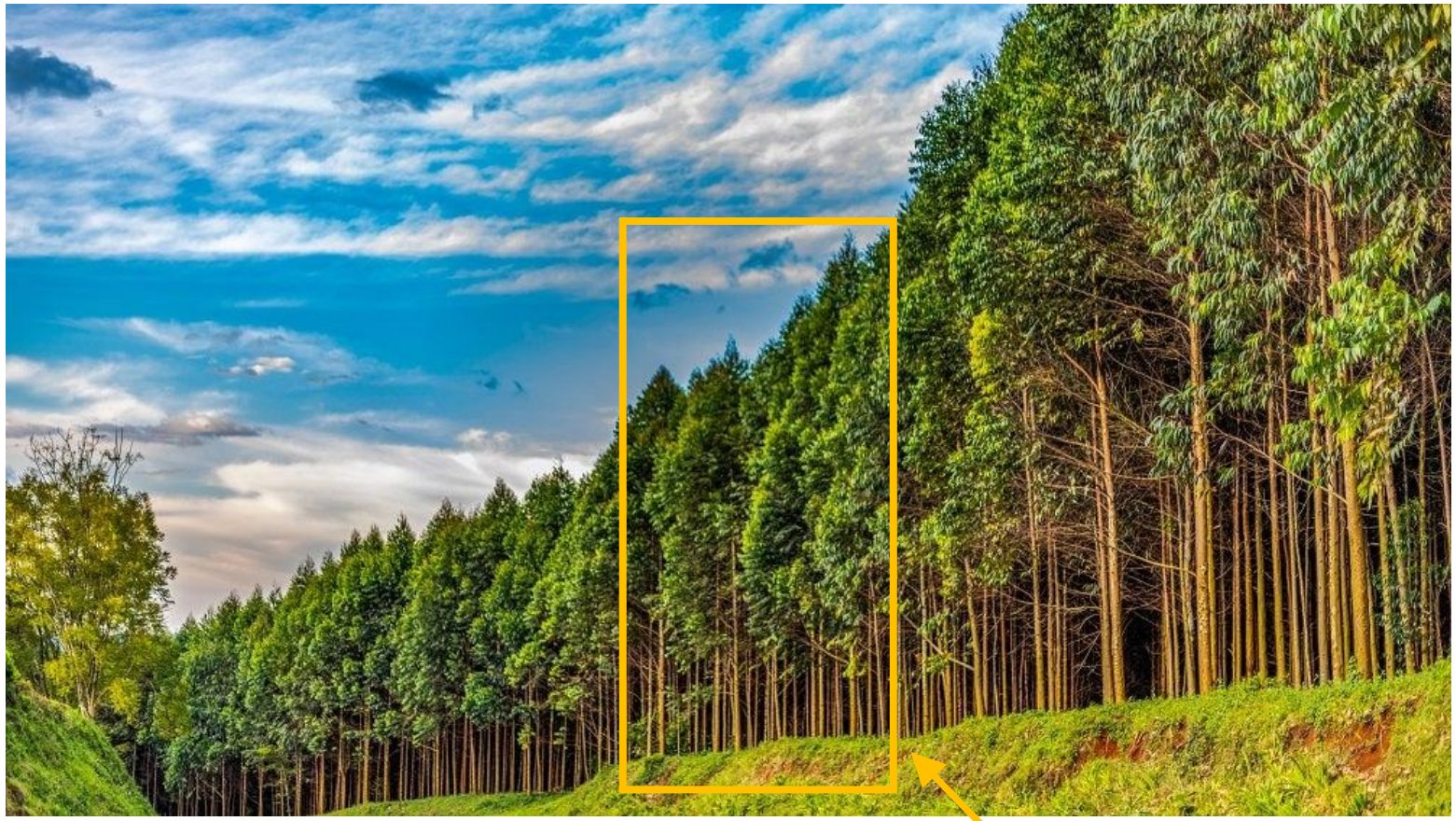
Introduction to the bootstrap method

Ciprian D Coman

Population vs. sample



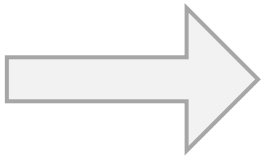
POPULATION
(the forest)

SAMPLE
(30 trees)

The central problem

What we want

The **sampling distribution** of a statistic
(e.g., the median, a correlation coefficient,
a model parameter, etc)



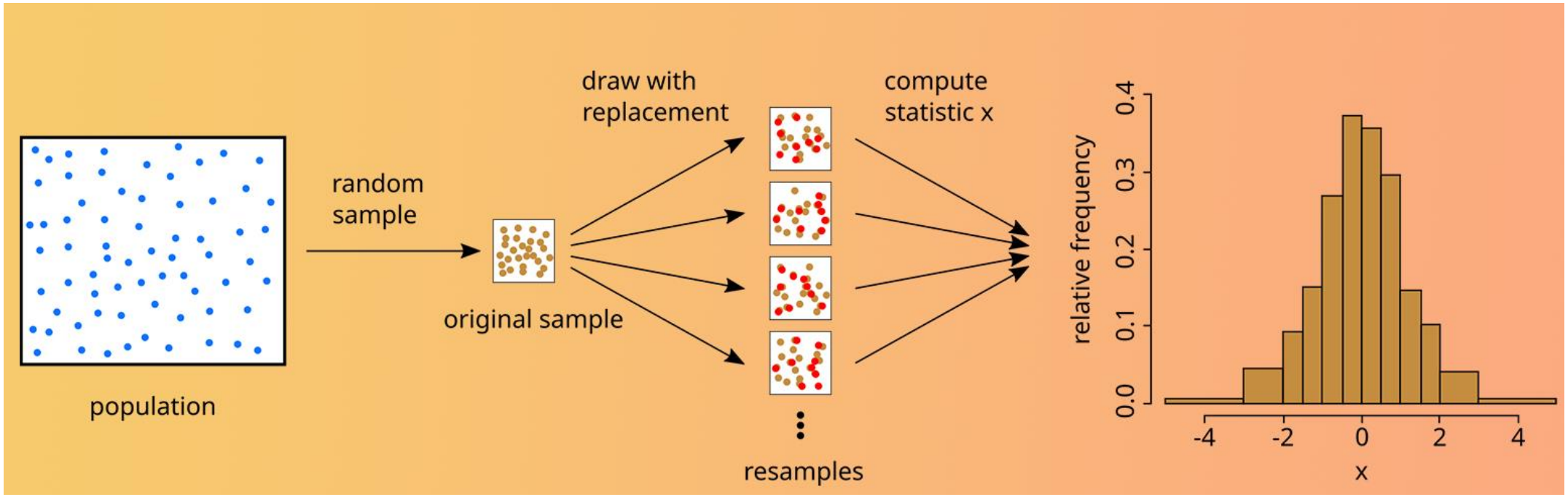
What we have…

A single observed sample of size n,
and usually **no easy analytical formula**

Bootstrapping: what is it?

- A **resampling procedure** that can be used to estimate the sampling distribution of almost any statistic, such as the mean, median, and regression coefficients

- Introduced by **Bradley Efron** in **1979**



How it works: the algorithm

1. **OBSERVE:** Collect your original sample $x_1, x_2, \dots, x_n$ of size n from the population

2. **RESAMPLE:** Draw n observations **with replacement** from the sample.
This is one bootstrap resample $x^* = \{x_1^*, x_2^*, \dots, x_n^*\}$

3. **COMPUTE:** Calculate your statistic of interest T^* from the resample (mean, median, etc)

4. **REPEAT:** Repeat steps 2-3 a large number of times B (typically, $B = 1{,}000 - 10{,}000$)

5. **SUMMARISE:** The empirical distribution of $T_1^*, T_2^*, \dots, T_B^*$ approximates the sampling distribution of T

Example

Suppose the **heights** of trees in a forest are normally distributed with mean $\mu = 18$ m and standard deviation $\sigma = 2$ m.

We observe 30 trees:

TREE →	1	2	3	4	5	6	7	8	9	10
HEIGHT →	14.5	15.0	15.3	15.8	16.0	16.2	16.5	16.7	16.9	17.1

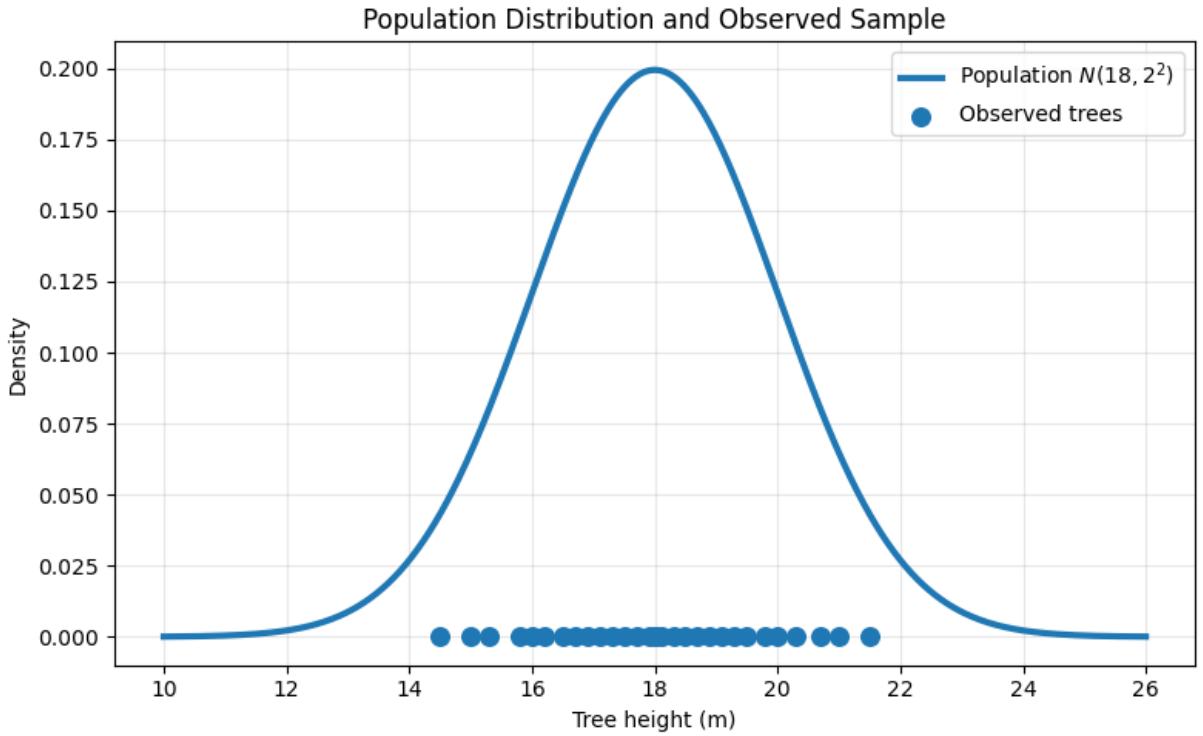
11	12	13	14	15	16	17	18	19	20
17.3	17.5	17.7	17.9	18.0	18.0	18.1	18.3	18.5	18.7

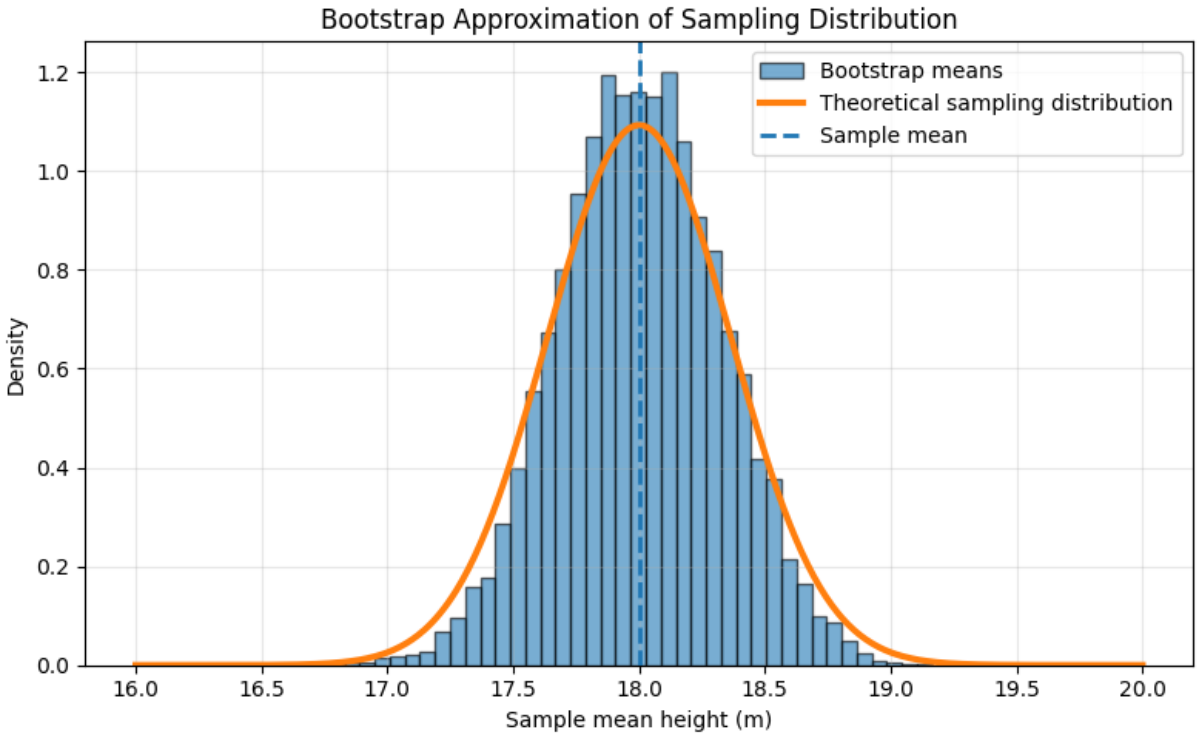
21	22	23	24	25	26	27	28	29	30
18.9	19.1	19.3	19.5	19.8	20.0	20.3	20.7	21.0	21.5

Example (cont'd)

sample mean = 18.00333
sample STD = 1.812692

theoretical SE = 0.3651483
bootstrap SE = 0.32208781





Cautionary example

Suppose the heights of trees in a forest are normally distributed with mean $\mu = 18$ m and standard deviation $\sigma = 2$ m.
We only observe 10 trees:

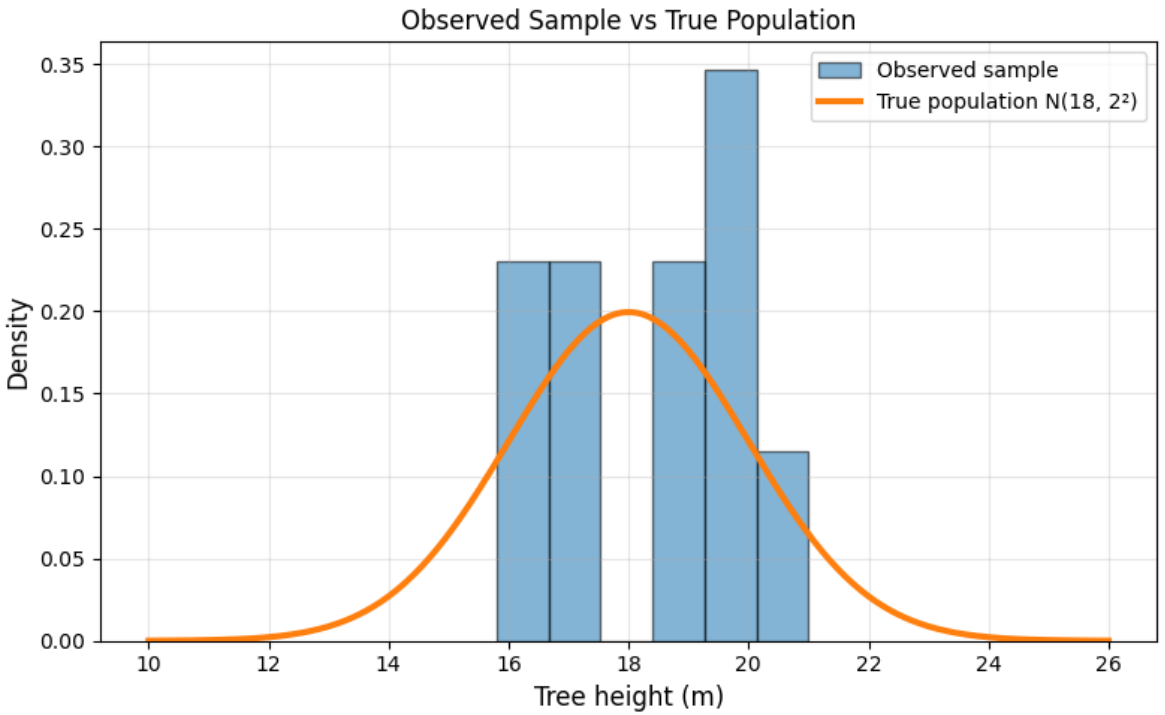
TREE	1	2	3	4	5	6	7	8	9	10
HEIGHT	16.2	17.5	19.3	20.1	18.4	15.8	21.0	17.1	18.9	19.7

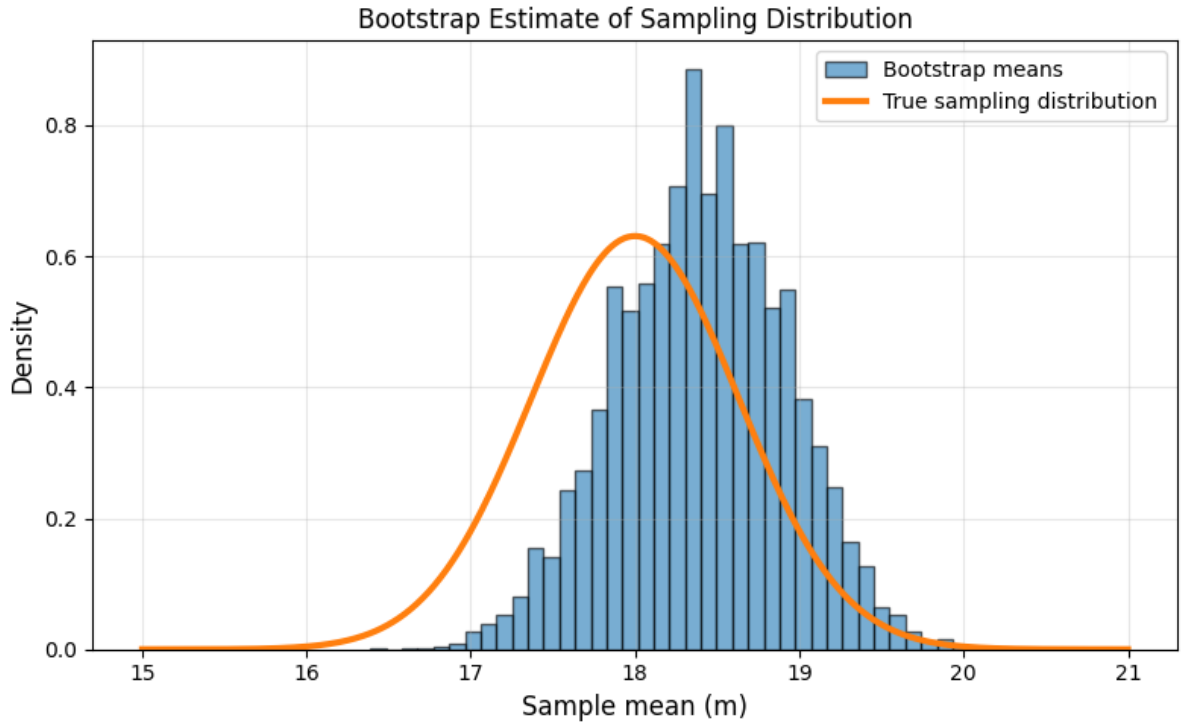
SAMPLE

$\bar{x} = 18.4$ m

Cautionary example (cont'd)

TREE	1	2	3	4	5	6	7	8	9	10
HEIGHT	16.2	17.5	19.3	20.1	18.4	15.8	21.0	17.1	18.9	19.7

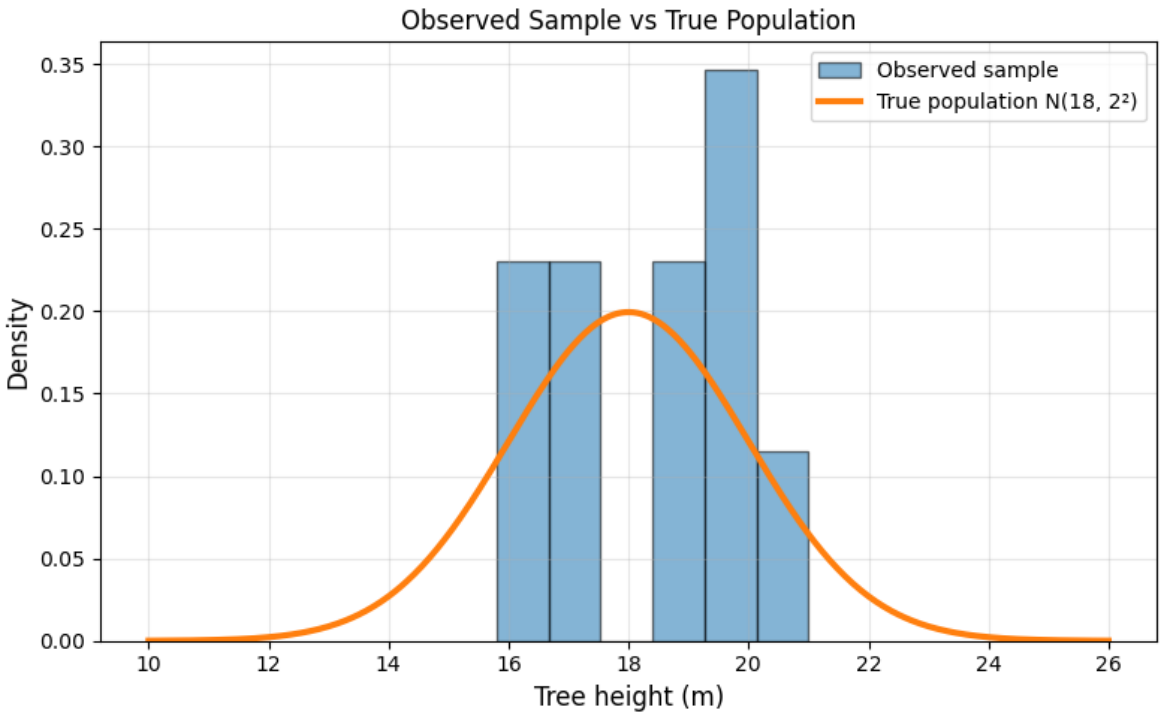


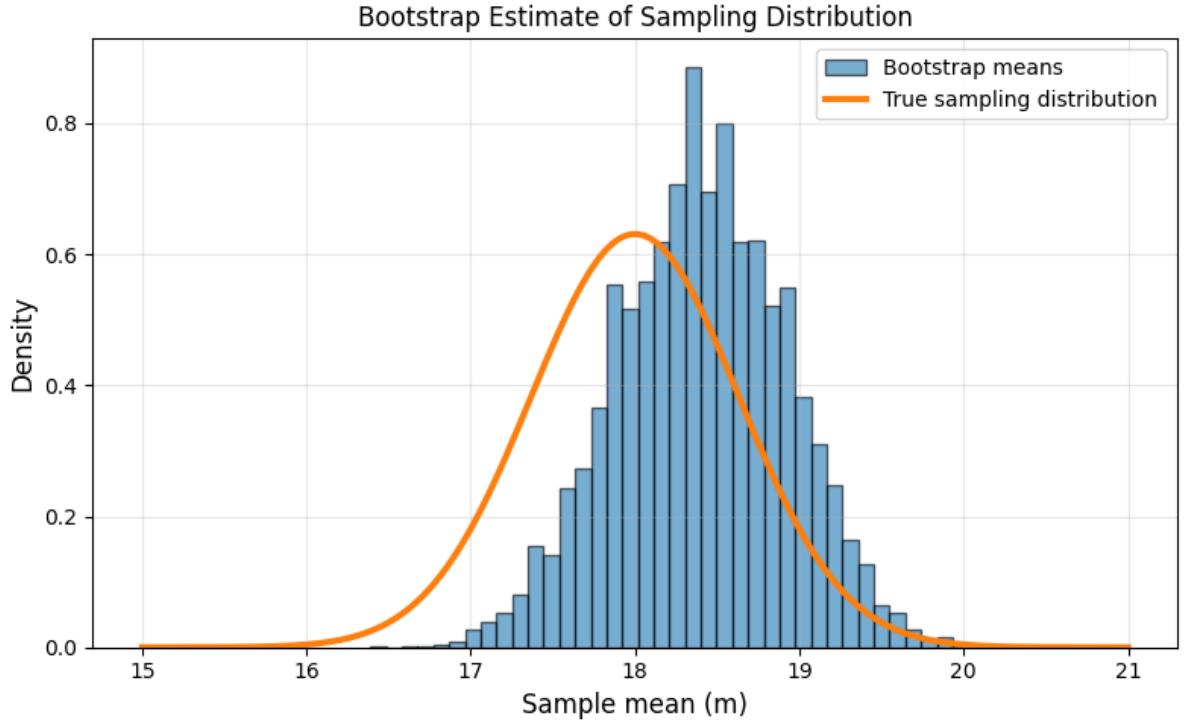


Cautionary example (cont'd)

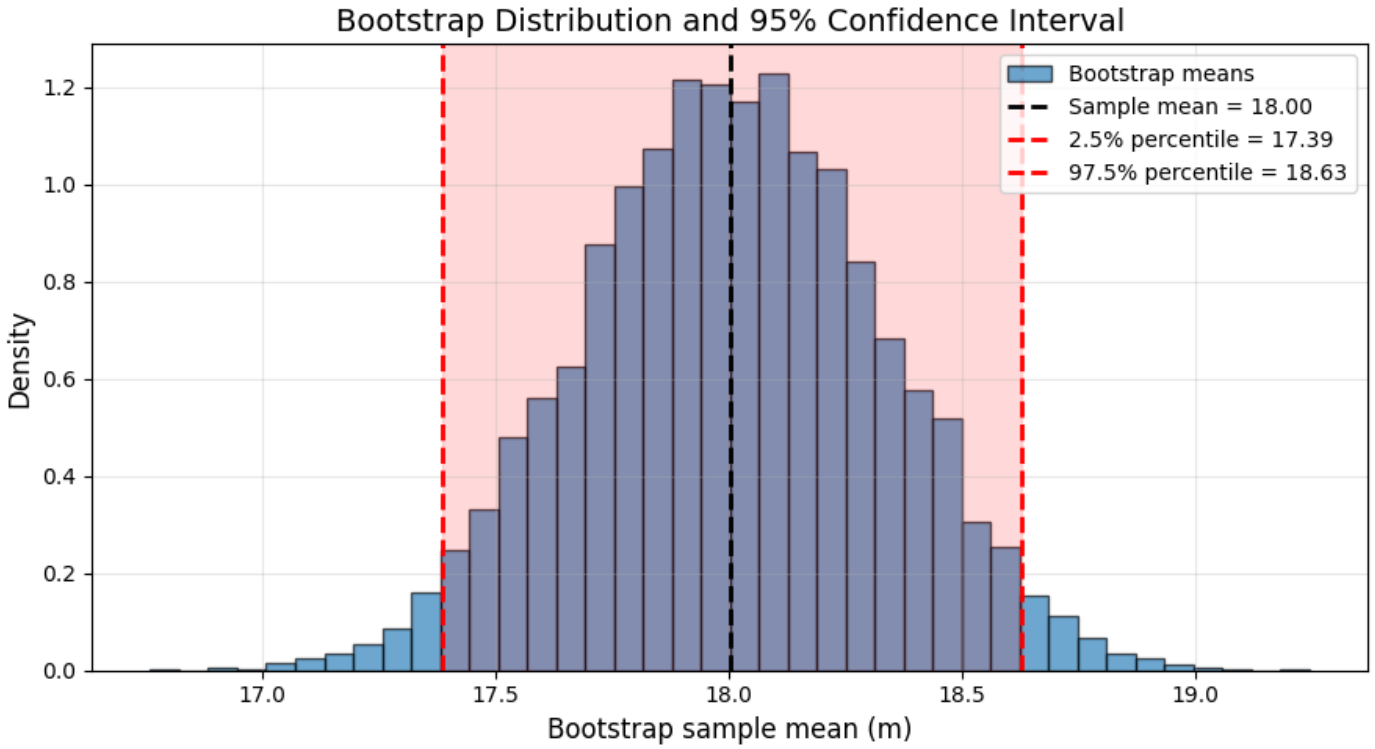
ATYPICAL SAMPLE

TREE	1	2	3	4	5	6	7	8	9	10
HEIGHT	16.2	17.5	19.3	20.1	18.4	15.8	21.0	17.1	18.9	19.7





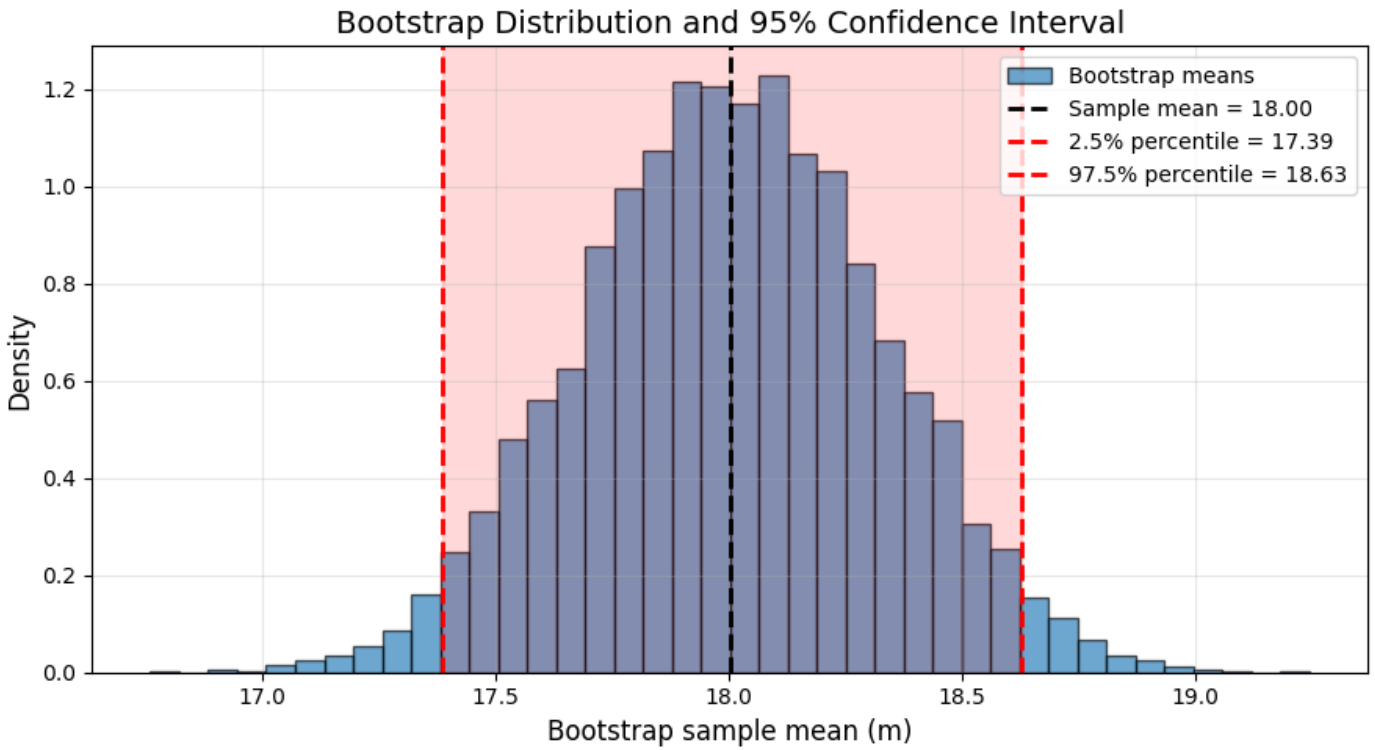
Bootstrap confidence intervals



95% bootstrap CI = [17.363, 18.637]

```
# ------------------------------------------------
# Bootstrap
# ------------------------------------------------

B = 10000

bootstrap_means = np.empty(B)

for k in range(B):

    sample_star = np.random.choice(
        heights,
        size=n,
        replace=True
    )

    bootstrap_means[k] = np.mean(sample_star)

# ------------------------------------------------
# Confidence interval
# ------------------------------------------------

lower, upper = np.percentile(
    bootstrap_means,
    [2.5, 97.5]
)

sample_mean = np.mean(heights)
```

Bootstrap confidence intervals



95% bootstrap CI = [17.363, 18.637]

classical formula: [17.28, 18.72]

```python
# ----------------------------------------
# Sample statistics
# ----------------------------------------

xbar = np.mean(heights)
s = np.std(heights, ddof=1)

print(f"Sample mean = {xbar:.3f}")
print(f"Sample standard deviation = {s:.3f}")

# ----------------------------------------
# Classical 95% t-confidence interval
# ----------------------------------------

alpha = 0.05

SE = s / np.sqrt(n)

tcrit = stats.t.ppf(
    1 - alpha/2,
    df=n-1
)

CI_t = (
    xbar - tcrit * SE,
    xbar + tcrit * SE
)

print("\nClassical t interval")
print(CI_t)
```

Key take-aways

- **Resampling with replacement** from the **observed sample** mimics drawing new samples from the population
- The **empirical distribution** of a bootstrap statistic approximates the **true sampling distribution**
- **Works for many statistics,** but relies on the observed sample being representative of the population