Nonparametric Bootstrap: rationale

Nonparametric Bootstrap : A statistical method that involves repeatedly resampling the observed data with replacement to generate a large number of bootstrap samples. These samples are used to estimate the sampling distribution of a statistic or to make inferences about population parameters without assuming a specific parametric distribution for the data.

1. Nonparametric bootstrap is advantageous for small sample sizes, where traditional parametric methods can be unreliable.
2. It provides a robust approach when data doesn't conform to specific distributional assumptions.
3. It is useful for estimating parameters, constructing confidence intervals, and conducting hypothesis tests without relying on distribution assumptions.
4. It is useful in scenarios involving complex data or when modeling assumptions are uncertain.

Nonparametric Bootstrap: introduction (1/2)

So suppose that we observe a sample $x_1, ..., x_n$ which are realizations of i.i.d. r.v. $X_1, ..., X_n$, with distribution L_x and with CDF F_x unknown.. The empirical Cummulative Distribution Function (eCDF) is defined as

$$F_n(x) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{x_i \leq x}$$

and is an estimator of the true CDF F_x. Moreover, we have that

$$E[F_n(x)] = F_x$$

and

$$var(F_n(x)) = \frac{1}{n}\Big((F_x)(1 - F_x)\Big)$$

Nonparametric Bootstrap: introduction (2/2)

Besides, from the Law of Large Numbers (LLN), we have that

$$F_n(x) \xrightarrow{a.s.} F_x \qquad \text{as } n \text{ goes to } \infty.$$

And from the Central Limit Theorem (CLT), we have that

$$F_n(x) \xrightarrow{L} N\left(F_x, \frac{1}{n}F_x(1 - F_x)\right) \qquad \text{as } n \text{ goes to } \infty.$$

The underlying idea of nonparametric bootstrap is that bootstrap samples can be generated from the $F_n(x)$ when F_x is unknow since $F_n(x)$ is a consistent estimator for F_x .

General algorithm

1. B independent samples $x_{b1}^*, ..., x_{bn}^*$, $b = 1, ..., B$ are drawn with replacement from the data $x_1, ..., x_n$. This is important, we resample with replacement from the data to obtain bootstrap samples.

2. each of the B independent samples, an estimate $\hat{\theta}_b^*$ is computed (the bootstrap replicates of $\hat{\theta}$).

3. The expectation of $\hat{\theta}$ (given $F_n(x)$) is estimated as follows:

$$E_{boot}[\hat{\theta}] = \frac{1}{B} \sum_{b=1}^{B} \hat{\theta}_b^*$$

4. The variance of $\hat{\theta}$ (given $F_n(x)$) is estimated as follows:

$$var_{boot}(\hat{\theta}) = \frac{1}{B-1} \sum_{b=1}^{B} \left(\hat{\theta}_b^* - \frac{1}{B} \sum_{b=1}^{B} \hat{\theta}_b^* \right)^2$$

Example

We observe a sample $x_1, ..., x_n = 8.26, 6.33, 10.4, 5.27, 5.35, 5.61, 6.12,$ $6.19, 5.2, 7.01, 8.74, 7.78, 7.02, 6, 6.5, 5.8, 5.12, 7.41, 6.52, 6.21, 12.28,$ $5.6, 5.38, 6.6, 8.74.$

Estimate the bias and the standard error of $\hat{\theta}$, the estimate of the coefficient of variation. The coefficient of variation cv is equal to

$$cv = sd(x)/mean(x)$$

Let's perform the analysis in R and Python.

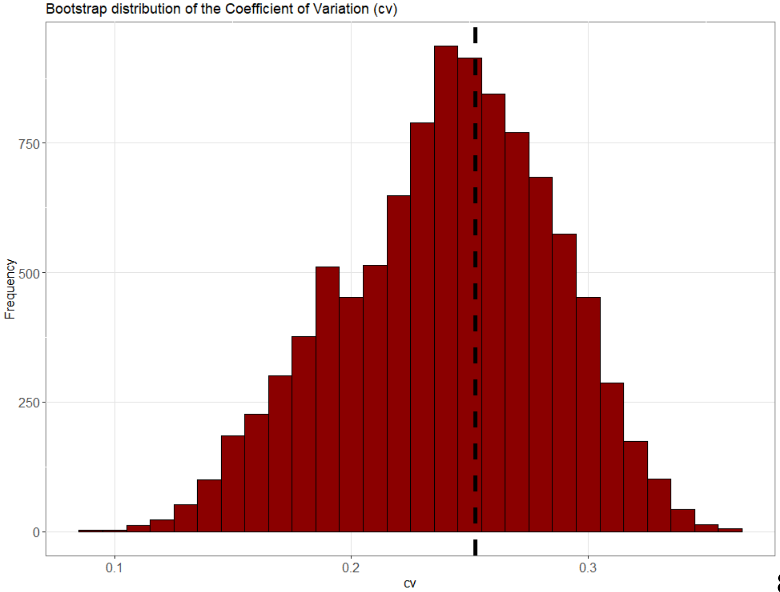
Example in R

```r
# dataset
x <- c(8.26, 6.33, 10.4, 5.27, 5.35, 5.61, 6.12, 6.19, 5.2, 7.01, 8.74, 7.78,
       7.02, 6, 6.5, 5.8, 5.12, 7.41, 6.52, 6.21, 12.28, 5.6, 5.38, 6.6, 8.74)

# coefficient of variation (CV)
cv <- sd(x) / mean(x)
cv # 0.2524712

# bootstrap
num_bootstraps <- 10000
bootstrap_cvs <- numeric(num_bootstraps)
set.seed(2023)
for (i in 1:num_bootstraps) {
  resample <- sample(x, replace = TRUE)
  bootstrap_cvs[i] <- sd(resample) / mean(resample)
}

# bias and standard error of the CV estimator
bias <- mean(bootstrap_cvs) - cv
bias # [1] -0.01216963
standard_error <- sd(bootstrap_cvs)
standard_error # [1] 0.04484109
```

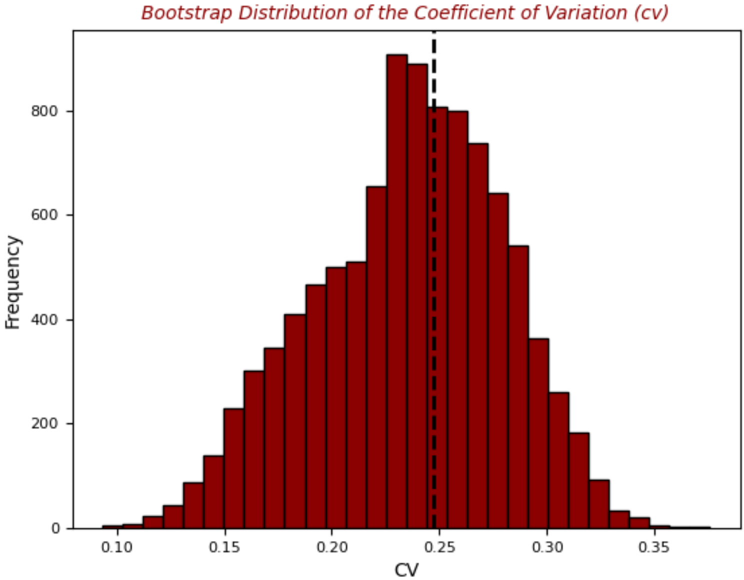
Example in Python

```python
import numpy as np
import pandas as pd

# dataset
x = np.array([8.26, 6.33, 10.4, 5.27, 5.35, 5.61, 6.12, 6.19, 5.2, 7.01, 8.74, 7.78,
              7.02, 6, 6.5, 5.8, 5.12, 7.41, 6.52, 6.21, 12.28, 5.6, 5.38, 6.6, 8.74])

# coefficient of variation (CV)
cv = np.std(x) / np.mean(x)
cv # 0.24737024423748963

# bootstrap
num_bootstraps = 10000
bootstrap_cvs = np.zeros(num_bootstraps)
np.random.seed(2023)
for i in range(num_bootstraps):
    resample = np.random.choice(x, size=len(x), replace=True)
    bootstrap_cvs[i] = np.std(resample) / np.mean(resample)

# bias and standard error of the CV estimator
bias = np.mean(bootstrap_cvs) - cv
bias #-0.011917596931186353
standard_error = np.std(bootstrap_cvs)
standard_error # 0.04395079547052894
```

Visualizing the distribution in R



Visualizing the distribution in Python



References

Rizzo, M. L. (2019), Statistical Computing with R, Second Edition, Chapman Hall/CRC, ISBN 9781466553330

The R Project for Statistical Computing:
https://www.r-project.org/

Python:
https://www.python.org/

course notes