

# Optimization and Simulation

## Statistical analysis and bootstrapping

Michel Bierlaire

Transport and Mobility Laboratory  
School of Architecture, Civil and Environmental Engineering  
Ecole Polytechnique Fédérale de Lausanne



**EPFL**

# Introduction

- The outputs of the simulator are random variables.
- Running the simulator provides one realization of these r.v.
- We have no access to the pdf or CDF of these r.v.
- Well... this is actually why we rely on simulation.
- How to derive statistics about a r.v. when only instances are known?
- How to measure the quality of this statistic?

## Sample mean and variance

- Consider  $X_1, \dots, X_n$  i.i.d. r.v.
- $E[X_i] = \mu, \text{Var}(X_i) = \sigma^2$ .

### The sample mean

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i$$

is an unbiased estimate of the population mean  $\mu$ , as  $E[\bar{X}] = \mu$ .

### The sample variance

$$S^2 = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2$$

is an unbiased estimator of the population variance  $\sigma^2$ , as  $E[S^2] = \sigma^2$ .  
(see proof: Ross, chapter 7)

# Sample mean and variance

## Recursive computation

1 Initialize  $\bar{X}_0 = 0$ ,  $S_1^2 = 0$ .

2 Update the mean

$$\bar{X}_{k+1} = \bar{X}_k + \frac{X_{k+1} - \bar{X}_k}{k+1}$$

3 Update the variance

$$S_{k+1}^2 = \left(1 - \frac{1}{k}\right) S_k^2 + (k+1)(\bar{X}_{k+1} - \bar{X}_k)^2.$$

# Mean Square Error

- Consider  $X_1, \dots, X_n$  i.i.d. r.v. with CDF  $F$ .
- Consider a parameter  $\theta(F)$  of the distribution (mean, quantile, mode, etc.)
- Consider  $\hat{\theta}(X_1, \dots, X_n)$  an estimator of  $\theta(F)$ .
- The Mean Square Error of the estimator is defined as

$$\text{MSE}(F) = E_F \left[ \left( \hat{\theta}(X_1, \dots, X_n) - \theta(F) \right)^2 \right],$$

where  $E_F$  emphasizes that the expectation is taken under the assumption that the r.v. all have distribution  $F$ .

- If  $F$  is unknown, it is not immediate to find an estimator of MSE.

## How many draws must be used?

- Let  $X$  a r.v. with mean  $\theta$  and variance  $\sigma^2$ .
- We want to estimate the mean  $\theta$  of the simulated distribution.
- The estimator used is the sample mean:  $\bar{X}$ .
- The mean square error is

$$E[(\bar{X} - \theta)^2] = \frac{\sigma^2}{n}$$

- The sample mean  $\bar{X}$  is normally distributed with mean  $\theta$  and variance  $\sigma^2/n$ .
- So we can stop generating data when  $\sigma/\sqrt{n}$  is small.
- $\sigma^2$  is approximated by the sample variance  $S$ .
- Law of large numbers: at least 100 draws (say) should be used.
- See Ross p. 121 for details.

# Mean Square Error

- Other indicators than the mean are desired.
- Theoretical results about the MSE cannot always be derived.
- Solution: rely on simulation.
- Method: bootstrapping.

# Empirical distribution function

- Consider  $X_1, \dots, X_n$  i.i.d. r.v. with CDF  $F$ .
- Consider a realization  $x_1, \dots, x_n$  of these r.v.
- The **empirical distribution function** is defined as

$$F_e(x) = \frac{1}{n} \sum_{i=1}^n I\{x_i \leq x\},$$

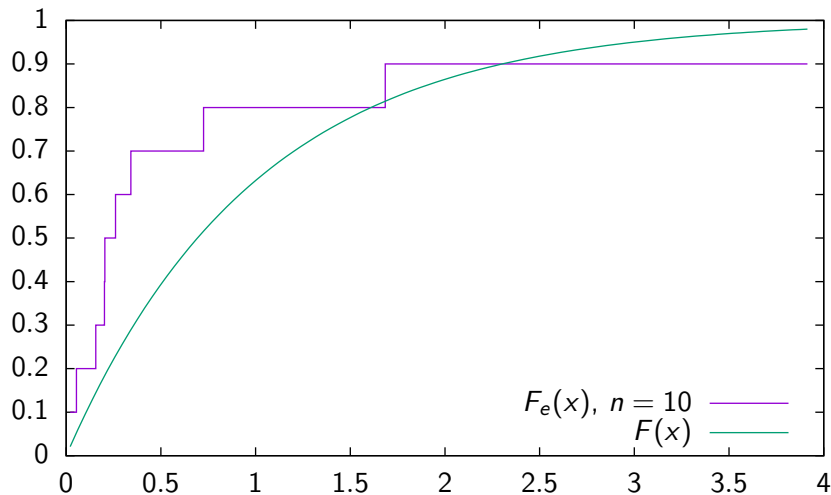
where

$$I\{x_i \leq x\} = \begin{cases} 1 & \text{if } x_i \leq x, \\ 0 & \text{otherwise.} \end{cases}$$

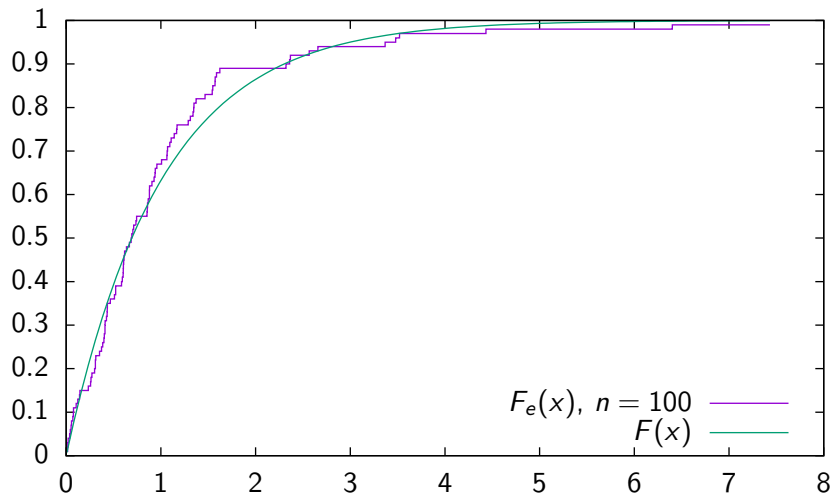
- CDF of a r.v. that can take any  $x_i$  with equal probability.



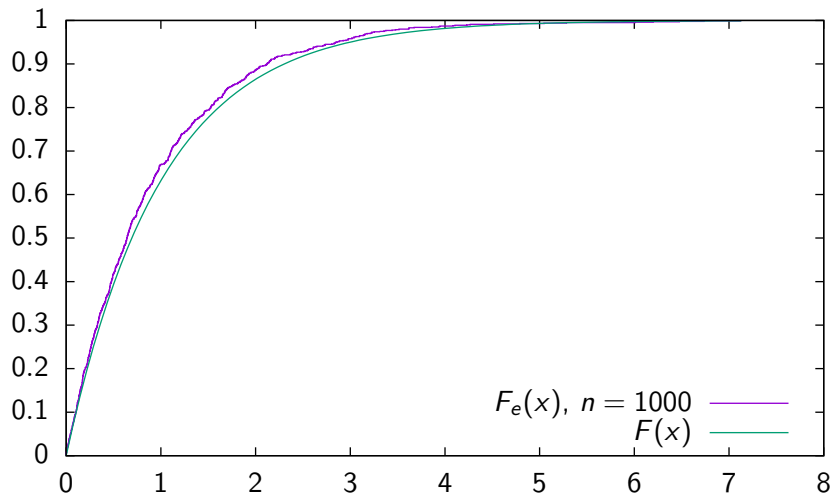
# Empirical CDF



# Empirical CDF



# Empirical CDF



## From reality to data

	Reality	Data
Random variable	$X$	$X_e$
CDF	$F$	$F_e$
True parameter	$\theta(F)$	$\theta(F_e)$
Sample	$X_1, \dots, X_n \sim F$	$X_1^e, \dots, X_n^e \sim F_e$
Estimate	$\hat{\theta}(X_1, \dots, X_n)$	$\hat{\theta}(X_1^e, \dots, X_n^e)$

# Mean Square Error

- We use the empirical distribution function  $F_e$
- We can approximate

$$\text{MSE}(F) = E_F \left[ \left( \hat{\theta}(X_1, \dots, X_n) - \theta(F) \right)^2 \right],$$

by

$$\text{MSE}(F_e) = E_{F_e} \left[ \left( \hat{\theta}(X_1^e, \dots, X_n^e) - \theta(F_e) \right)^2 \right],$$

- $\theta(F_e)$  can be computed directly from the data (mean, variance, etc.)

# Mean Square Error

- We want to compute

$$\text{MSE}(F_e) = E_{F_e} \left[ \left( \widehat{\theta}(X_1^e, \dots, X_n^e) - \theta(F_e) \right)^2 \right],$$

- $X_i^e$  are r.v. that can take any  $x_i$  with equal probability.
- Therefore,

$$\text{MSE}(F_e) = \frac{1}{n^n} \sum_{i_1=1}^n \cdots \sum_{i_n=1}^n \left[ \left( \widehat{\theta}(x_{i_1}, \dots, x_{i_n}) - \theta(F_e) \right)^2 \right],$$

- Clearly impossible to compute when  $n$  is large.
- Solution: simulation.

# Bootstrapping

- For  $r = 1, \dots, R$
- Draw  $x_1^r, \dots, x_n^r$  from  $F_e$ , that is draw from the data:
  - ① Let  $s$  be a draw from  $U[0, 1]$
  - ② Set  $j = \text{floor}(ns)$ .
  - ③ Return  $x_j$ .

- Compute

$$M_r = \left( \hat{\theta}(x_1^r, \dots, x_n^r) - \theta(F_e) \right)^2,$$

- Estimate of  $\text{MSE}(F_e)$  and, therefore, of  $\text{MSE}(F)$ :

$$\frac{1}{R} \sum_{r=1}^R M_r$$

- Typical value for  $R$ : 100.

# Bootstrap: simple example

- Data: 0.636, -0.643, 0.183, -1.67, 0.462
- Mean= -0.206
- $MSE = E[(\bar{X} - \theta)^2] = S^2/n = 0.1817$

$r$						$\hat{\theta}$	$\theta(F_e)$	MSE
1	-0.643	-0.643	-0.643	0.462	0.462	-0.201	-0.206	2.544e-05
2	-0.643	0.183	0.636	0.636	0.636	0.2896	-0.206	0.2456
3	-1.67	-1.67	0.183	0.462	0.636	-0.411	-0.206	0.04204
4	-1.67	-0.643	0.183	0.183	0.636	-0.2617	-0.206	0.003105
5	-0.643	0.462	0.462	0.636	0.636	0.3105	-0.206	0.2667
6	-1.67	-1.67	0.183	0.183	0.183	-0.5573	-0.206	0.1234
7	-0.643	0.183	0.183	0.462	0.636	0.1642	-0.206	0.137
8	-1.67	-1.67	-0.643	0.183	0.183	-0.7225	-0.206	0.2667
9	0.183	0.462	0.462	0.636	0.636	0.4756	-0.206	0.4646
10	-0.643	0.183	0.183	0.462	0.636	0.1642	-0.206	0.137
								0.1686



## Python code

```
def bootstrap(data):
    n = len(data)
    b = []
    for i in range(0,n):
        r = random()
        index = int(n*r)
        b.append(data[index])
    return b

def percentile(dataSorted,p):
    i = max(int(round(p * len(dataSorted) + 0.5)), 2)
    return dataSorted[i-2]
```

## MSE for the percentile: Python code

```
N = 10000
data = drawNormal(N)
data.sort()
theP = 0.975
quantile = float(percentile(data,theP))
print(f'{theP}% quantile={quantile:.4f}')
R=100
sum = 0.0
for l in range (0,R):
    b = bootstrap(data)
    b.sort()
    q = float(percentile(b,theP))
    sum += (quantile-q)**2
print(f'MSE: {sum/R:.4f} sqrt(MSE): {np.sqrt(sum/(R)):.4f}')
```

# Results

N=10000

0.975% quantile=1.9472

MSE: 0.0009 sqrt(MSE): 0.0303

N=1000

0.975% quantile=1.8808

MSE: 0.0098 sqrt(MSE): 0.0988

N=100

0.975% quantile=1.4078

MSE: 0.0300 sqrt(MSE): 0.1732

# Summary

- The number of draws is determined by the required precision.
- In some cases, the precision is derived from theoretical results.
- If not, rely on bootstrapping.
- Idea: use simulation to estimate the Mean Square Error.

## Appendix: MSE for the mean

- Consider  $X_1, \dots, X_n$  i.i.d. r.v.
- Denote  $\theta = E[X_i]$  and  $\sigma^2 = \text{Var}(X_i)$ .
- Consider  $\bar{X} = \sum_{i=1}^n X_i/n$ .
- $E[\bar{X}] = \sum_{i=1}^n E[X_i]/n = \theta$ .
- MSE:

$$\begin{aligned} E[(\bar{X} - \theta)^2] &= \text{Var } \bar{X} \\ &= \text{Var} \left( \sum_{i=1}^n X_i/n \right) \\ &= \sum_{i=1}^n \text{Var}(X_i)/n^2 \\ &= \sigma^2/n. \end{aligned}$$