

## Definition

The **bootstrap** is a simulation method for estimating variation of point estimates and computing confidence intervals

Let  $T_n = t(X_1, X_2, \dots, X_n)$  be a statistic, with  $X_1 \sim F(\cdot)$

Suppose we want to know  $Var_F(T_n)$

Basic bootstrap idea has the following three steps:

- - ➊ Estimate  $F(\cdot)$  from a random sample  $X_1, X_2, \dots, X_n$  to obtain empirical distribution function  $F_n(\cdot)$
  - ➋ Simulate from  $F_n(\cdot)$  to obtain  $b$  *bootstrap samples*
  - ➌ Approximate  $Var_F(T_n)$  with  $Var_{F_n}(T_n)$
- Real  $F \implies X_1, X_2, \dots, X_n \circlearrowleft \implies T_n = t(X_1, X_2, \dots, X_n)$   
 Bootstrap  $F_n \implies X_1^*, X_2^*, \dots, X_n^* \implies T_n^* = t(X_1^*, X_2^*, \dots, X_n^*)$
- - To estimate  $F_n(\cdot)$  we put  $1/n$  probability over each data point from  $X_1, X_2, \dots, X_n$
  - Therefore, drawing an observation from  $F_n(\cdot)$  is equivalent to drawing one point at random from the original random sample
  - **Important:** we always draw at random with replacement
  - **Important:** the size of a single bootstrap sample equals the size of the original random sample  $n$

## Bootstrap Example

Suppose we roll a 6-sided die  $n = 10$  times and get the following data, written in increasing order:

$$1, 1, 2, 3, 3, 3, 4, 5, 6, 6.$$



Imagine writing these values on 10 pieces of paper, putting them in a hat and drawing one at random. Then, e.g., probability of drawing a 3 is  $3/10$  and of 4 is  $1/10$ . We can put the full *empirical distribution*  $P_{10}$  in a probability table:

- | $x$         | 1    | 2    | 3    | 4    | 5    | 6    |
|-------------|------|------|------|------|------|------|
| $P_{10}(x)$ | 2/10 | 1/10 | 3/10 | 1/10 | 1/10 | 2/10 |

Notice the difference between the true distribution  $P$  and the empirical distribution  $P_{10}$ .

$x$	1	2	3	4	5	6
$P(x)$	1/6	1/6	1/6	1/6	1/6	1/6
$P_{10}(x)$	2/10	1/10	3/10	1/10	1/10	2/10

Remark: By the WLLN we know that  $P_n(x) \xrightarrow{P} P(x), \forall x, n \rightarrow \infty$ . Since in practice we often do not know  $P$  we resample from  $P_n$ :

- ① We draw from  $P_n$  with replacements! We draw a piece of paper from the hat at random, then we **put back** this piece of paper in the hat.
- ② We draw in this way  $n$  times to obtain a single bootstrap sample 
- ③ We repeat steps 1 and 2  $b$  times to obtain  $b$  bootstrap samples

For example, for  $b = 6$  we may obtain:

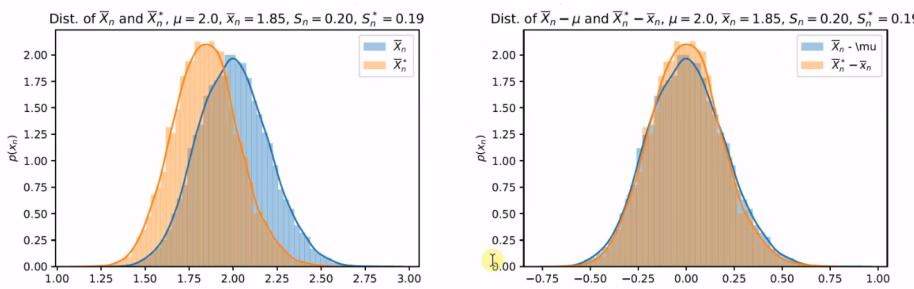
$$\begin{aligned}
 X_1, X_2, \dots, X_{10} &: 1, 1, 2, 3, 3, 3, 4, 5, 6, 6. \\
 X_{1,1}^*, X_{2,1}^*, \dots, X_{10,1}^* &: 1, 1, 1, 3, 3, 4, 4, 5, 5, 6. \\
 X_{1,2}^*, X_{2,2}^*, \dots, X_{10,2}^* &: 1, 2, 3, 3, 3, 3, 4, 4, 5, 6. \\
 X_{1,3}^*, X_{2,3}^*, \dots, X_{10,3}^* &: 2, 2, 3, 3, 3, 3, 5, 5, 5, 6. \\
 X_{1,4}^*, X_{2,4}^*, \dots, X_{10,4}^* &: 1, 2, 3, 4, 4, 4, 4, 6, 6, 6. \\
 X_{1,5}^*, X_{2,5}^*, \dots, X_{10,5}^* &: 1, 1, 2, 3, 3, 3, 4, 4, 5, 6. \\
 X_{1,6}^*, X_{2,6}^*, \dots, X_{10,6}^* &: 1, 2, 2, 3, 3, 3, 4, 6, 6, 6.
 \end{aligned}$$

Algorithm

- ① Estimate  $F_n$  from  $X_1, X_2, \dots, X_n$  by putting  $1/n$  probability over each data point (this is the maximum likelihood estimator for  $F$ )
- ② Draw  $X_1^*, X_2^*, \dots, X_n^* \sim F_n$  
- ③ Compute  $T_{n,i}^* = t(X_1^*, X_2^*, \dots, X_n^*)$
- ④ Repeat steps 2 and 3  $b$  times
- ⑤ Approximate variability of  $T_n$  by the variability of  $T_n^*$

Why does it work

- Distribution of  $T_n$  is centered at  $\theta$  (the parameter that  $T_n$  estimates)
- Likewise, the distribution of  $T_n^*$  is centered at  $t_n$  (the concrete realization of  $T_n$  for the original sample)
- If  $t_n$  and  $\theta$  differ significantly than the two distributions are also significantly different
- But the two distributions of the variation around the center are approximately equal:  $T_n - \theta$  and  $T_n^* - t_n$
- Alternatively, the bootstrap works because
  - ①  $F_n \approx F$
  - ② The amounts of variation in the estimates with  $F_n$  and  $F$  are similar
- 



[[Konfidenzintervall]]

- Let us define two new r.v.  $\delta_n = T_n - \theta$  and  $\delta_n^* = T_n^* - t_n$
- Bootstrap principle states that  $\delta_n \approx \delta_n^*$
- If we would know the distribution of  $\delta_n$  then  $1 - \alpha$  confidence intervals for  $T_n$  are:

$$\begin{aligned} P(\delta_{n_{\alpha/2}} < T_n - \theta < \delta_{n_{1-\alpha/2}}) &= 1 - \alpha \\ P(\delta_{n_{\alpha/2}} - T_n < -\theta < \delta_{n_{1-\alpha/2}} - T_n) &= 1 - \alpha \\ P(T_n - \delta_{n_{1-\alpha/2}} < \theta < T_n - \delta_{n_{\alpha/2}}) &= 1 - \alpha \end{aligned}$$

- Since we do not know the distribution of  $\delta_n$  but we can compute the distribution of  $\delta_n^*$  we replace the former with the latter to obtain *bootstrap confidence intervals*:

- $P(T_n - \delta_{n_{1-\alpha/2}}^* < \theta < T_n - \delta_{n_{\alpha/2}}^*) = 1 - \alpha$

Confidence Interval Example

Suppose  $n = 4$  radar guns are set up along a stretch of road to catch people driving over the speed limit. For a car passing at speed  $\mu$  four readings are (45.71, 47.41, 40.95, 50.65). Compute a random interval that covers the true unknown car speed  $\mu$  with probability of 0.95.

### Bootstrap confidence intervals

```
b = 10000
x_star = np.random.choice(x, (n, b))
x_star_bar = np.mean(x_star, axis=0)
delta_star = x_star_bar - x_bar
print('(% bootstrap confidence interval for mu: (:5.2f), (:5.2f))'.format((1 - alpha) * 100,
    x_bar - np.quantile(delta_star, 1 - alpha/2),
    x_bar - np.quantile(delta_star, alpha/2)))
95.0% bootstrap confidence interval for mu: (42.95, 49.80)
```



## Alternative Statistics

- bootstrap also capable of estimating distribution of almost any [[Statistik]]
- examples
  - skewness

### Bootstrap Confidence Intervals for Arbitrary Statistics

Example. The skewness is a measure of asymmetry of a distribution. A normal distribution, which is symmetric, has skewness 0. Estimate the confidence interval for the skewness of game sales in North America using the Game Sales dataset from Kaggle: <https://www.kaggle.com/gregorut/videogamesales>.

```
alpha = 0.05
name, x, eu, jp, other = ds.game_sales(count=-1)
n = len(x)
s = stats.skew(x)
print('Sample skewness: {:.4f}'.format(s))

b = 1000
x_star = np.random.choice(x, (n, b))
s_star = stats.skew(x_star, axis=0)
delta_star = s_star - s
print('(% bootstrap confidence interval for skewness: (:4.2f), (:4.2f))'.format((1 - alpha) * 100,
    s - np.quantile(delta_star, 1 - alpha/2), s - np.quantile(delta_star, alpha/2)))
Column names are Rank, Name, Platform, Year, Genre, Publisher, NA_Sales, EU_Sales, JP_Sales, Other_Sales, Global_Sales
Processed 16599 lines.
Sample skewness: 18.80
95.0% bootstrap confidence interval for skewness: (13.67, 26.99)
```



- bioequivalence

This example is from Efron and Tibshirani (inventors of bootstrap), An Introduction to the Bootstrap. When drug companies introduce new medications, they need to show bioequivalence, i.e., that the new drug is not substantially different than the current one. Here are data on eight subjects who used hormone infusing drug. Each subject received three treatments: placebo, old, new. Let  $X = \text{old-placebo}$  and  $Y = \text{new-old}$ . The bioequivalence is given if  $|\theta| \leq 0.2$ , where:

$$\theta = \frac{E_F(Y)}{E_F(X)}$$



Answer the bioequivalence question with bootstrap. The data is given in the table on the next slide.

placebo	old	new
9243	17649	16449
9671	12013	14614
11792	19979	17274
13357	21816	23798
9055	13850	12560
6290	9806	10157
12412	17208	16570
18806	29044	26325

\*

```

alpha = 0.05
placebo, old, new = ds.hormone_drug()
placebo = np.array(placebo)
old = np.array(old)
new = np.array(new)
bio = np.mean(new - old) / np.mean(old - placebo)
print('Sample bioequivalence: ', bio)
n = len(placebo)
x = range(n)

b = 1000
x_star = np.random.choice(x, (n, b))
bio_star = np.mean(new[x_star] - old[x_star], axis=0) / np.mean(old[x_star] - placebo[x_star], axis=0)
delta_star = bio_star - bio
print('()% bootstrap confidence interval for bioequivalence: ({:5.3f}, {:5.3f})'.format((1 - alpha) * 100,
    bio - np.quantile(delta_star, 1 - alpha/2),
    bio - np.quantile(delta_star, alpha/2)))

```

Sample bioequivalence: -0.07130609590256017  
95.0% bootstrap confidence interval for bioequivalence: (-0.295, 0.083)

We obtain  $(-0.291, 0.094)$  as a 95% bootstrap confidence interval,  $(-0.2, 0.2)$  is not contained in it and hence we have not demonstrated the bioequivalence.