

Table of Contents

- 2 Sampling Definitions
- 11 Unbiased Estimators
- 23 Standard Error
- 29 Bootstrap: Sample Mean
- 40 Bootstrap: Sample Variance
- 57 Bootstrap: p-values

19: Sampling and the Bootstrap

Jerry Cain
February 23, 2024



[Lecture Discussion on Ed](#)

Sampling definitions

- Population (모집단) : 조사의 대상이 되는 모든 개체 값의 집합
- Population parameter (모수) : 모집단을 조사하여 얻을 수 있는 통계적 특성치
- Sample (표본) : 샘플링된 모집단의 부분집합

Motivating example

You want to know the true mean and variance of happiness in Bhutan.

- But you can't ask everyone.
- You poll 200 random people.
- Your data looks like this:

$$\text{Happiness} = \{72, 85, 79, 91, 68, \dots, 71\}$$

- The mean of all these numbers is 83.

Is this the true mean happiness of all Bhutanese people?

*we really mean
the average happiness
of the entire
population*



Population

population might be $N=160,000$
(population of Bhutan is 771,000 as
of 2021)



This is a **population**.

Sample



A **sample** is selected from a population.

Sample



A **sample** is selected from a population.

Reasonable Questions Starting Out

e.g. interview, or analyze or otherwise observe.

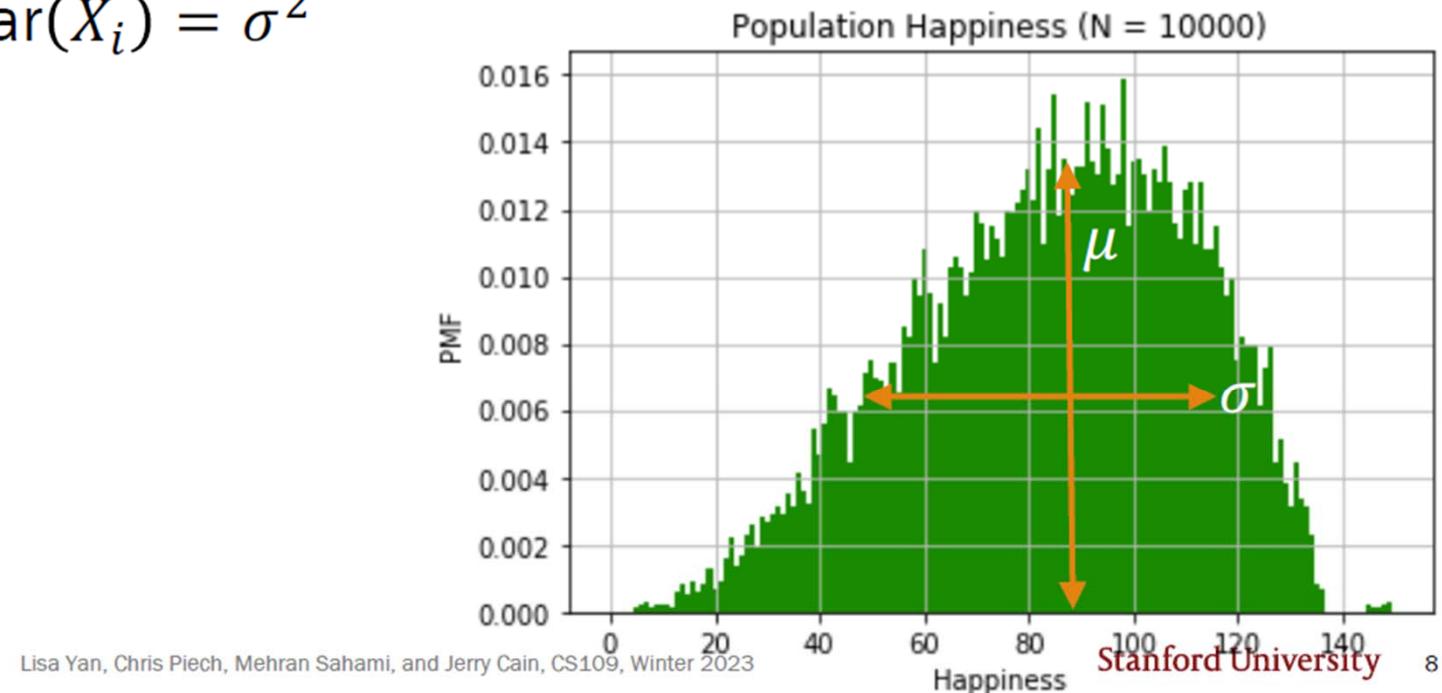
1. In situations where we can't observe the entire population, what can we safely conclude by polling a sample drawn from that population?
by safe, we mean reliably, i.e., backed by data and scientific method
2. How large does your sample need to be before your conclusions are trustworthy, and how do we express confidence with any conclusions we draw?
3. Are there alternative ways to infer population statistics without polling entire populations?

A sample, mathematically

Consider n random variables X_1, X_2, \dots, X_n .

The sequence X_1, X_2, \dots, X_n is a **sample** from distribution F if:

- X_i are all independent and identically distributed (iid)
- X_i all have same distribution function F (the **underlying distribution**), where $E[X_i] = \mu$, $\text{Var}(X_i) = \sigma^2$



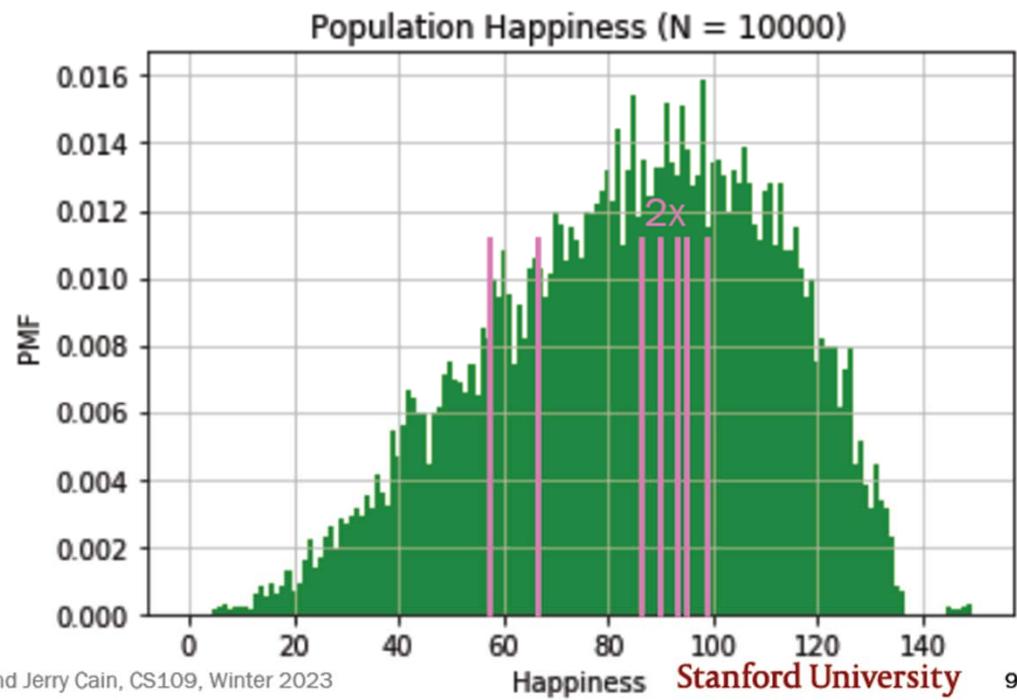
A sample, mathematically

A sample of **size** 8:

$$(X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8)$$

The **realization** of a sample of size 8:

$$(59, 87, 94, 99, 87, 78, 69, 91)$$



A single sample



A happy
Bhutanese person

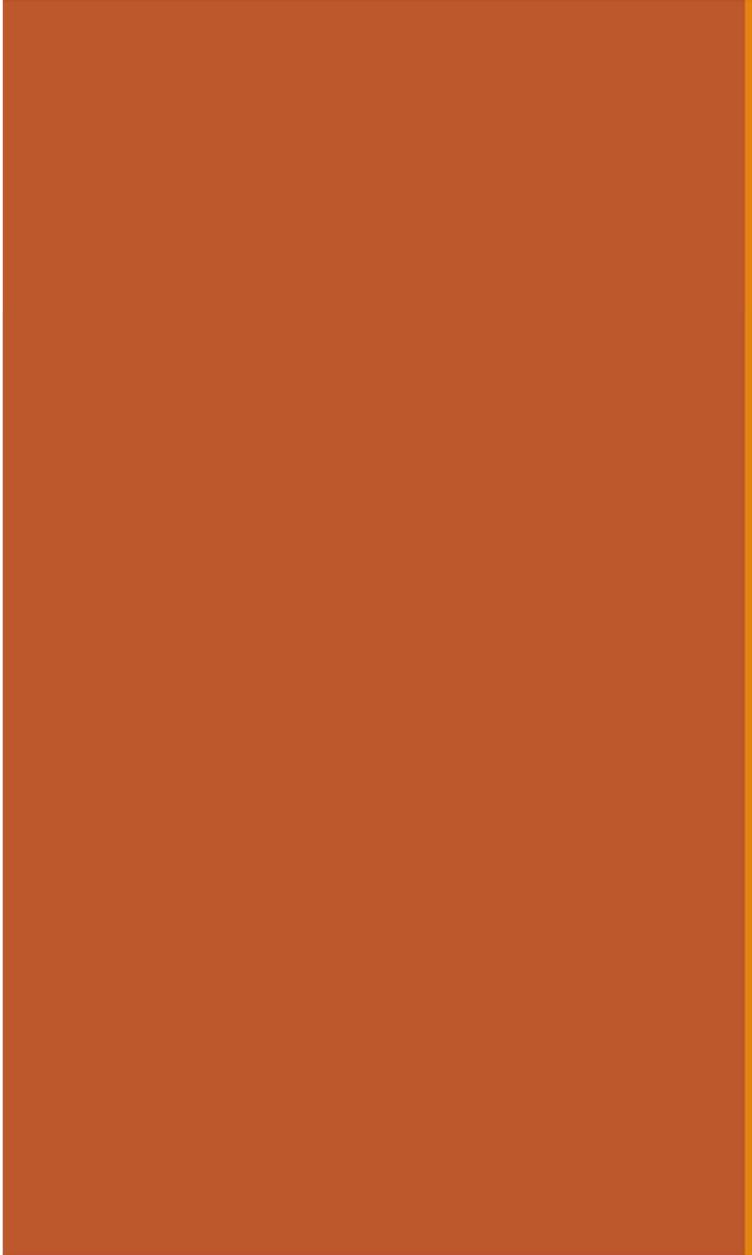
If we had a distribution F of our entire population, we could compute exact statistics about happiness.

the graph from prior slide pretend we do know the population distribution F , though in general we wouldn't

But we only have 200 people—or rather, a sample.

Today: If we only have a single sample, *perhaps of size 200*

- How do we report **estimated statistics**?
 - We're careful to call them **estimated mean** and **estimated variance**, since they're **based on samples** (i.e., experiments)
- How do we report **estimated errors** on these estimates?
- How do we perform something called **hypothesis testing**? Oh, and what is it?



Unbiased estimators

➤ **Bias** : 추정된 파라미터가 실제 파라미터와 다른 정도를 나타내는 스칼라 값임.

모델의 파라미터 θ 를 추정했을 때,

- Bias의 값이 크다면, 추정된 파라미터가 실제 파라미터와는 멀리 떨어져 있음을 나타냄.
- Bias 값이 작은 경우가 실제 파라미터와 값이 비슷할 것이고 대체적으로 Bias값이 0인 경우가 좋음 [출처: <https://jrc-park.tistory.com/267>]

$$E(\hat{\theta}) = \theta + bias(\theta)$$

- **Unbiased Estimator** : 파라미터 추정 평균에 대해서 bias 값이 0인 경우, **bias = 0**

- **Biased Estimator** : 파라미터 추정 평균의 bias값이 0이 아닌 경우, **bias ≠ 0**

예를 들어서, Gaussian 분포에서 반복적으로 x_1, x_2, \dots, x_N 값을 뽑아서, 첫 번째 원소로 Gaussian의 평균으로 추정하는 것은 Unbiased Estimator 임

A single sample



A happy
Bhutanese person

If we had a distribution F of our entire population, we could compute exact statistics about happiness. *But again, we generally do not.*

But we only have 200 people (a sample).

These population-level statistics are unknown:

- μ , the **population mean**
- σ^2 , the **population variance**

A single sample

If we had a distribution F of our entire population, we could compute exact statistics about happiness.



A happy
Bhutanese person

But we only have 200 people (a sample).

- From these 200 people, what is our best estimate of the **population mean** and the **population variance**?
- How exactly do we define best estimate?

Estimating the population mean



1. What is our best estimate of μ , the **mean happiness** of Bhutanese people?

If we only have (X_1, X_2, \dots, X_n) :

The best estimate of μ is the **sample mean**:

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

\bar{X} is an unbiased estimator of the population mean μ .

$$E[\bar{X}] = \mu$$

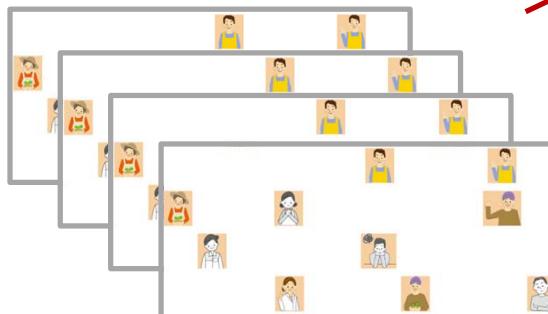
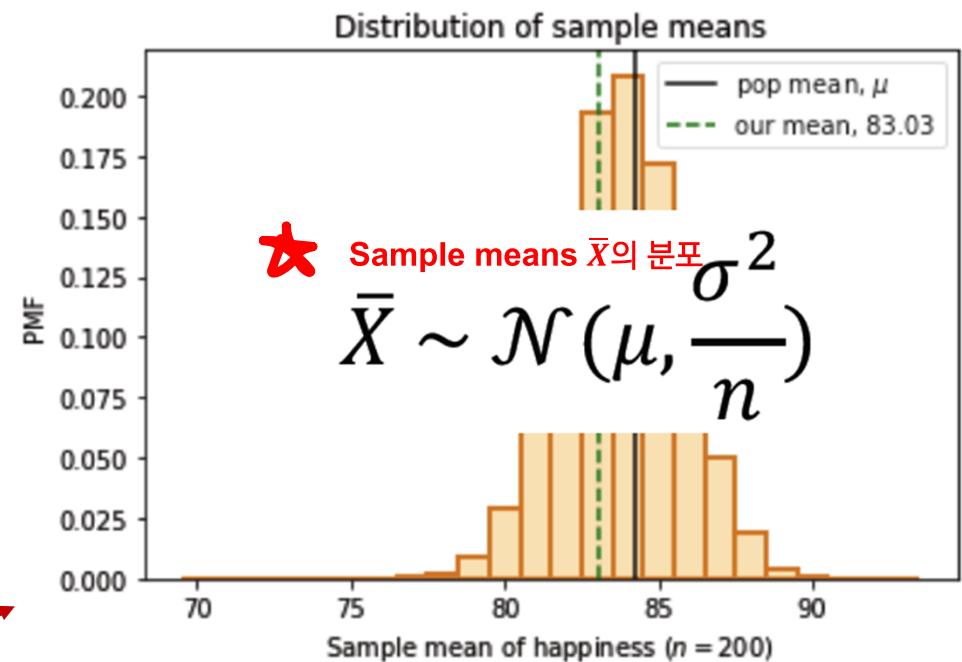
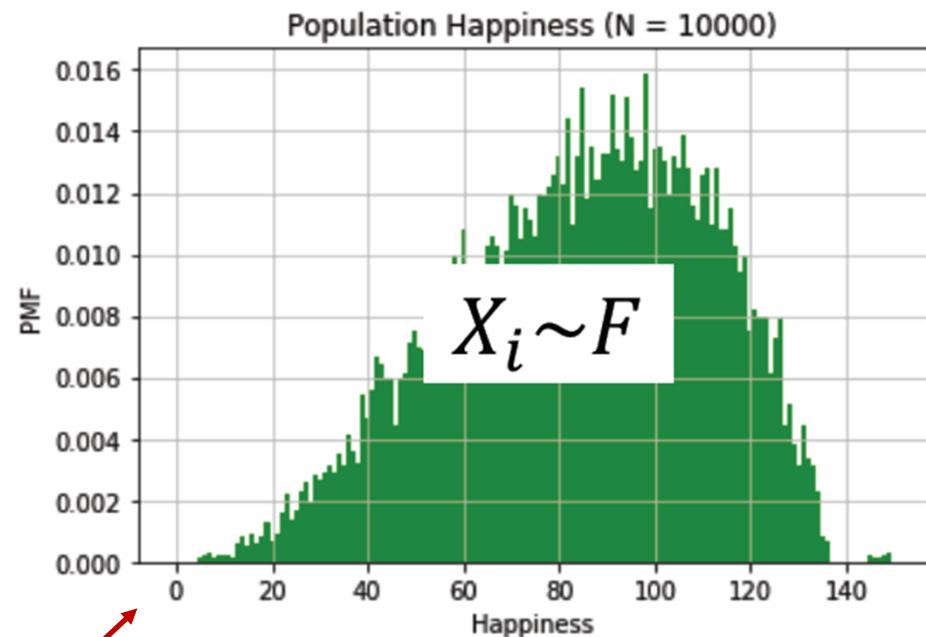
Intuition: By the CLT,

$$\bar{X} \sim \mathcal{N}\left(\mu, \frac{\sigma^2}{n}\right)$$

Central Limit Theorem (CLT) : 동일한 확률분포를 가진 독립 확률 변수 n 개의 평균의 분포는 n 이 적당히 크다면 정규분포에 가까워짐

- If we could take multiple samples of size n :
1. For each sample, compute sample mean
 2. On average, we would get the population mean

Sample mean



Even if we can't report μ , we can report our sample mean 83.03, which is an unbiased estimate of μ .

Estimating the population variance



2. What is σ^2 , the **variance of happiness** of Bhutanese people?

If we knew the entire population (x_1, x_2, \dots, x_N) :

population
variance

$$\sigma^2 = E[(X - \mu)^2] = \frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2$$

population mean

If we only have one sample: (X_1, X_2, \dots, X_n) : sample mean

sample
variance

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

- S^2 은 σ^2 에 대한 정확한 추정치를 제시해주는 Variance인, "Unbiased estimator"임
- $\frac{1}{n-1}$ 은 Unbiased estimator으로 만들기 위한 보정 계수임 (S^2 은 자유도 $n-1$ 의 chi-square 분포에 맞음) $(n-1)S^2/\sigma^2$

[출처] <https://tamref.com/22>

Intuition about the sample variance, S^2

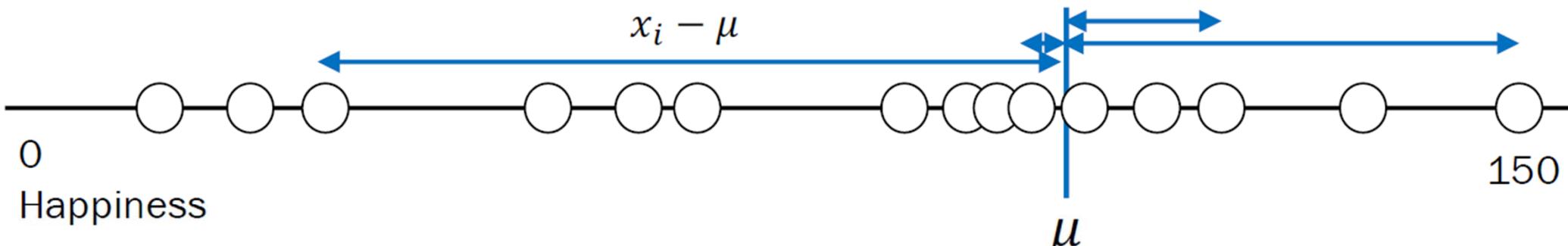
Actual, σ^2

population variance

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2$$

population mean \downarrow

you have perfect information if you have access to the full population because that's pretty much everything there is to know.



Population size, N

Calculating population statistics exactly requires us knowing all N datapoints.

Intuition about the sample variance, S^2

Actual, σ^2

population variance

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2$$

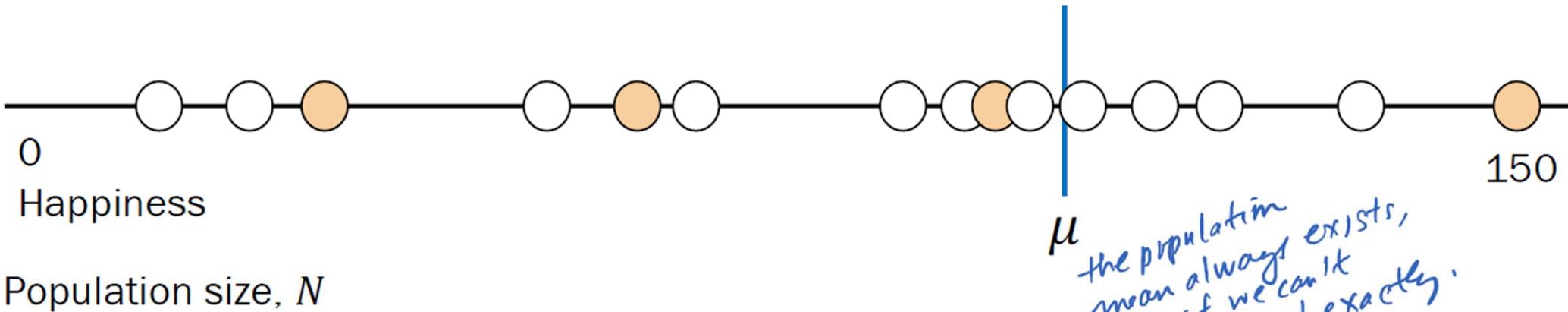
population mean
↓

Estimate, S^2

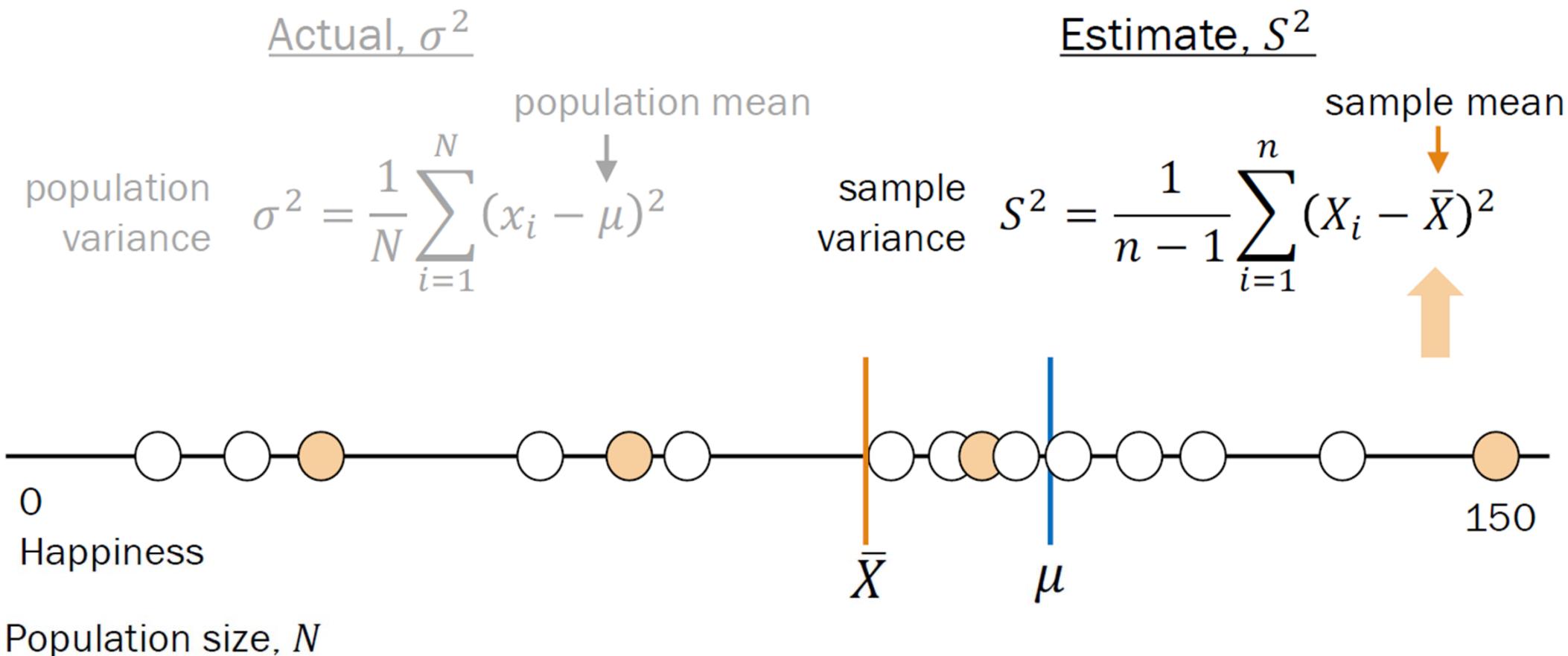
sample variance

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

sample mean
↓



Intuition about the sample variance, S^2



Intuition about the sample variance, S^2

Actual, σ^2

population variance

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2$$

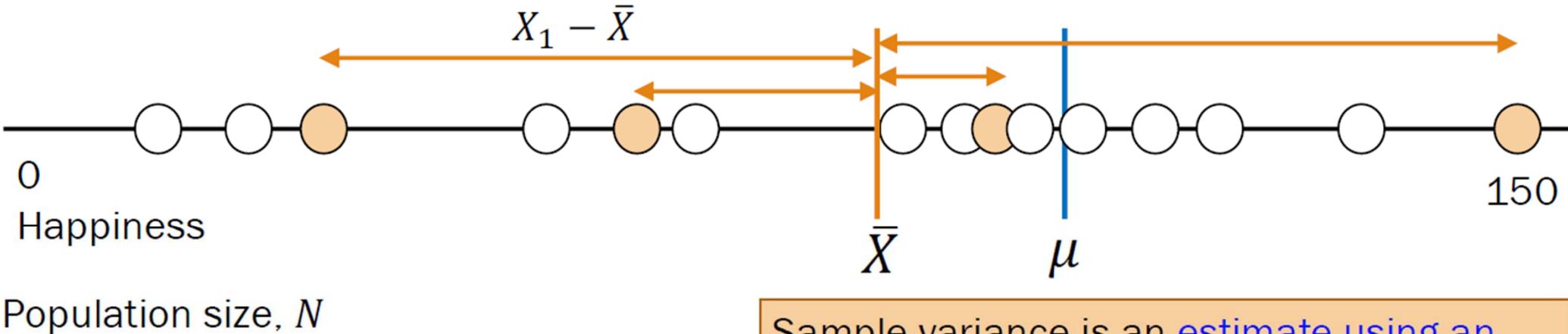
population mean \downarrow

Estimate, S^2

sample variance

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

sample mean \downarrow



Sample variance is an estimate using an estimate, so it requires additional scaling.

Estimating the population variance



2. What is σ^2 , the **variance of happiness** of Bhutanese people?

If we only have a sample, (X_1, X_2, \dots, X_n) :

The best estimate of σ^2 is the **sample variance**:

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

$\frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2$ is also an estimator, but it's biased, because it's generally ^{too} low

S^2 is an unbiased estimator of the population variance, σ^2 .

$$E[S^2] = \sigma^2$$

Proof that S^2 is unbiased (just for reference)

$$E[S^2] = \sigma^2$$

$$E[S^2] = E\left[\frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2\right] \Rightarrow (n-1)E[S^2] = E\left[\sum_{i=1}^n (X_i - \bar{X})^2\right]$$

$$(n-1)E[S^2] = E\left[\sum_{i=1}^n ((X_i - \mu) + (\mu - \bar{X}))^2\right] \quad (\text{introduce } \mu - \mu)$$

$$= E\left[\sum_{i=1}^n (X_i - \mu)^2 + \sum_{i=1}^n (\mu - \bar{X})^2 + 2 \sum_{i=1}^n (X_i - \mu)(\mu - \bar{X})\right] \rightarrow 2(\mu - \bar{X}) \sum_{i=1}^n (X_i - \mu)$$

$$= E\left[\sum_{i=1}^n (X_i - \mu)^2 + n(\mu - \bar{X})^2 - 2n(\mu - \bar{X})^2\right] \rightarrow 2(\mu - \bar{X}) \left(\sum_{i=1}^n X_i - n\mu\right)$$

$$= E\left[\sum_{i=1}^n (X_i - \mu)^2 - n(\mu - \bar{X})^2\right] = \sum_{i=1}^n E[(X_i - \mu)]^2 - nE[(\bar{X} - \mu)^2]$$

$$= n\sigma^2 - n\text{Var}(\bar{X}) = n\sigma^2 - n\frac{\sigma^2}{n} = n\sigma^2 - \sigma^2 = (n-1)\sigma^2$$

Therefore $E[S^2] = \sigma^2$



Standard error

Estimating population statistics

A particular outcome

1. Collect a sample, X_1, X_2, \dots, X_n .

$$(72, 85, 79, 79, 91, 68, \dots, 71)$$

$$n = 200$$

2. Compute **sample mean**, $\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i$.

$$85 - 83$$

$$79 - 83$$

$$\bar{X} = 83$$

$$72 - 83$$

$$79 - 83$$

3. Compute sample deviation, $X_i - \bar{X}$.

$$(-11, 2, -4, -4, 8, -15, \dots, -12)$$

4. Compute **sample variance**, $S^2 = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2$.

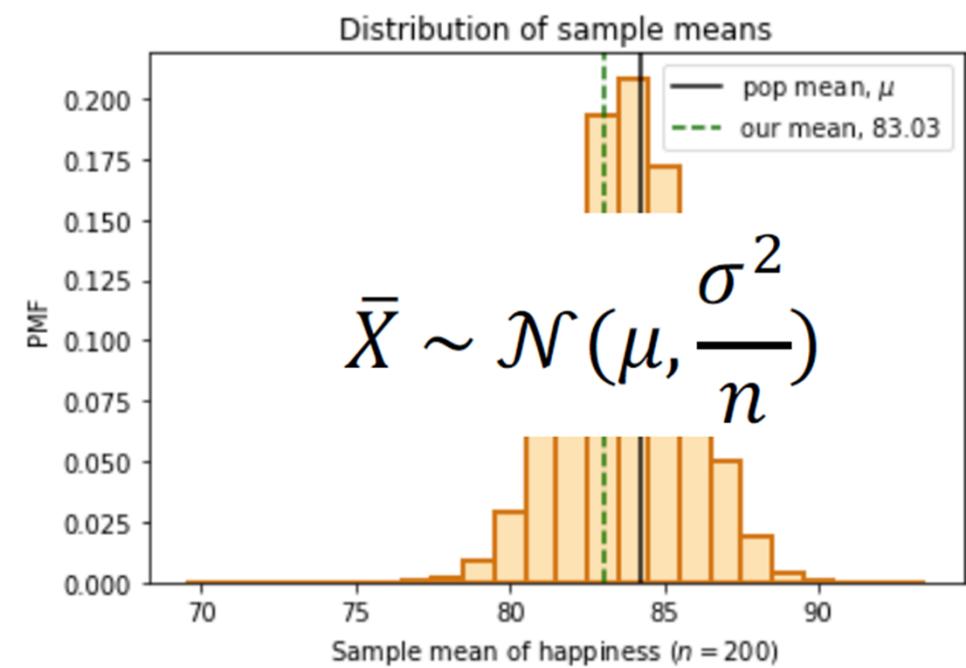
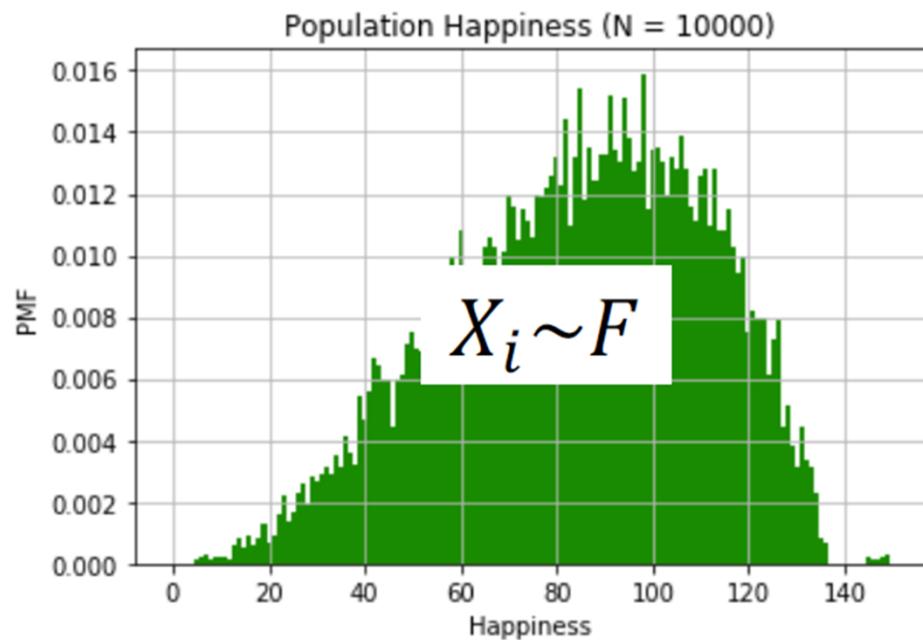
$$S^2 = 793$$

how close is \bar{X} to μ ?

how close is S^2 to σ^2 ?

How close are our estimates \bar{X} and S^2 ?

Sample mean



- $\text{Var}(\bar{X})$ is a measure of how close \bar{X} is to μ .
- How do we estimate $\text{Var}(\bar{X})$?

if $\text{Var}(\bar{X})$ is large, we're not very confident that \bar{X} is all that close to μ .
if $\text{Var}(\bar{X})$ is small, we're more confident.

How close is our estimate \bar{X} to μ ?

$$E[\bar{X}] = \mu$$

$$\text{Var}(\bar{X}) = \frac{\sigma^2}{n}$$

$$\text{SD}(\bar{x}) = \sqrt{\text{Var}(\bar{X})} = \sqrt{\frac{\sigma^2}{n}}$$

but remember!
we don't know σ^2 .
that's a population-level statistic

def The standard error of the mean is an estimate of the standard deviation of \bar{X} .

$$SE = \sqrt{\frac{S^2}{n}}$$

Intuition:

- S^2 is an unbiased estimate of σ^2
- S^2/n is an unbiased estimate of $\sigma^2/n = \text{Var}(\bar{X})$
- $\sqrt{S^2/n}$ can estimate $\sqrt{\text{Var}(\bar{X})}$
(somewhat biased, but best we can do)

$E[SE] < \text{SD}(\bar{x})$
less than because of the bias
More info on bias of standard error: [wikipedia](#)

Standard error of the mean

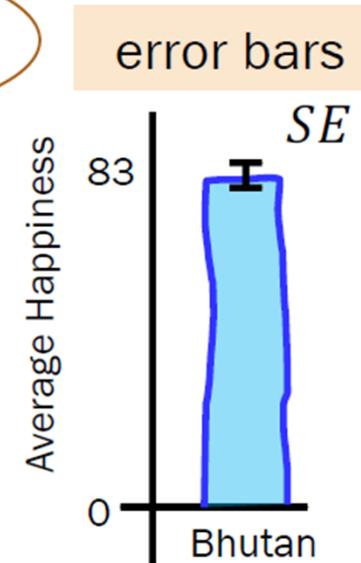
1. Mean happiness:

Claim: The average happiness of Bhutan is 83, with a standard error of 1.99.

Closed form: $SE = \sqrt{\frac{S^2}{n}}$

this is our estimate of how close we are *to μ*

this is our best estimate of μ



앞에 예시

$$\bar{X} = 83$$

$$S^2 = 793$$

$$n = 200$$

These 2 statistics give a sense of how \bar{X} —that is, the sample mean random variable—behaves.

Standard error of variance?

1. Mean happiness:

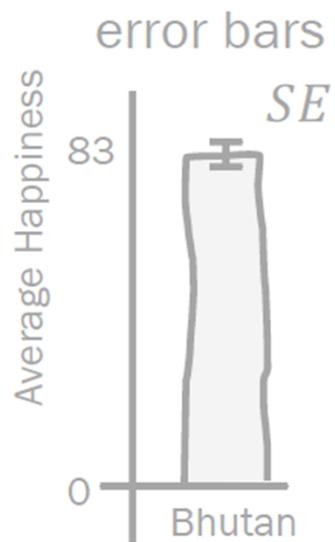
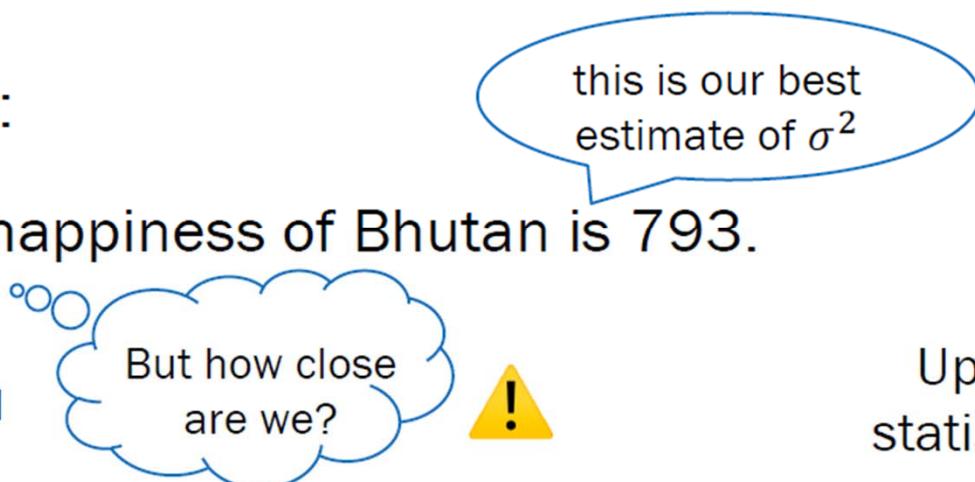
Claim: The average happiness of Bhutan is 83, with a standard error of 1.99.

Closed form: $SE = \sqrt{\frac{s^2}{n}}$

2. Variance of happiness:

Claim: The variance of happiness of Bhutan is 793.

Closed form: Not covered in CS109



Up next: Compute statistics with code!

Bootstrap: Sample mean

Bootstrap

The Bootstrap:

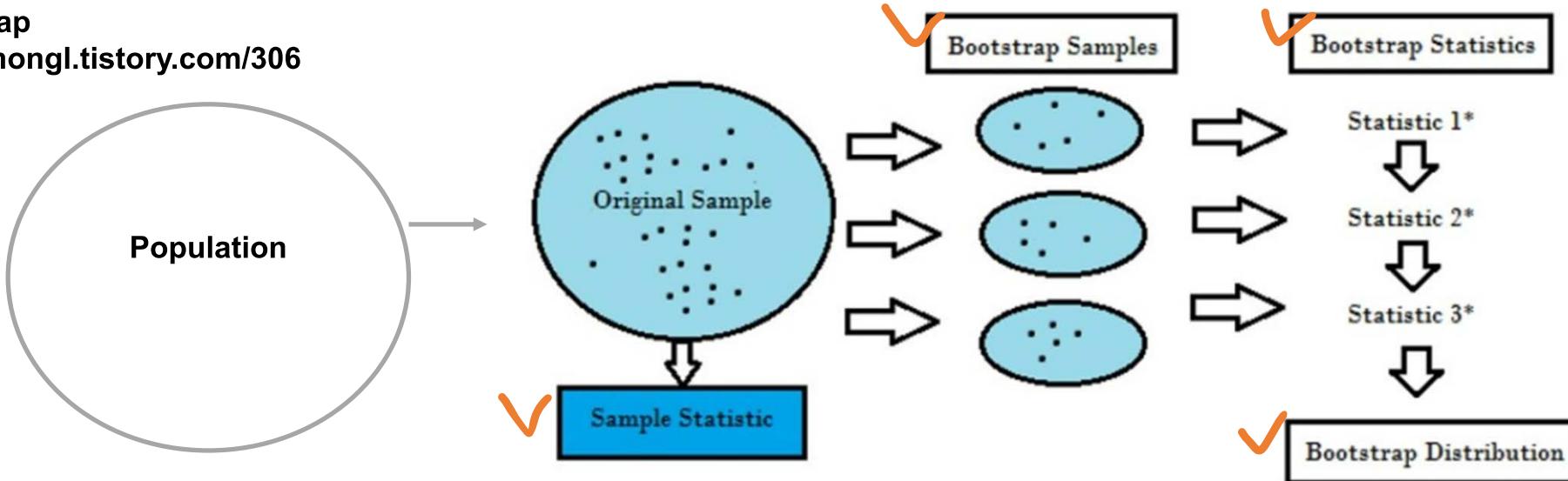
Probability for Computer Scientists

Bootstrap sampling은 하나의 sample data set을 n개의 sample data set을 가지도록 하며, 이를 통해 data의 variance를 상당히 잘 근사 할 수 있는 결과를 볼 수 있음

[An Introduction to the Bootstrap Method | by Lorna Yen | Towards Data Science](#)

<https://towardsdatascience.com/an-introduction-to-the-bootstrap-method-58bcb51b4d60>

Bootstrap
<https://hongl.tistory.com/306>



부트스트랩 알고리즘 : 현재 가지고 있는 샘플에서

- 1) 여러 번의 복원 추출을 시행하고 (**bootstrap sample**),
- 2) 복원 추출해서 얻은 샘플들의 평균을 추정하여 (**bootstrap statistics**),
- 3) 모집단(**population, universe**)의 분포를 추정하는 (**bootstrap distribution**) 방식으로 데이터 x_i 가 가우스 분포를 따른다고 가정하지 않고 복원 추출로부터 리샘플링된 샘플의 평균 $\mu = (1/N)\sum x_i$ 으로부터 모수의 분포를 (평균, 오차) 추정하겠다는 것임

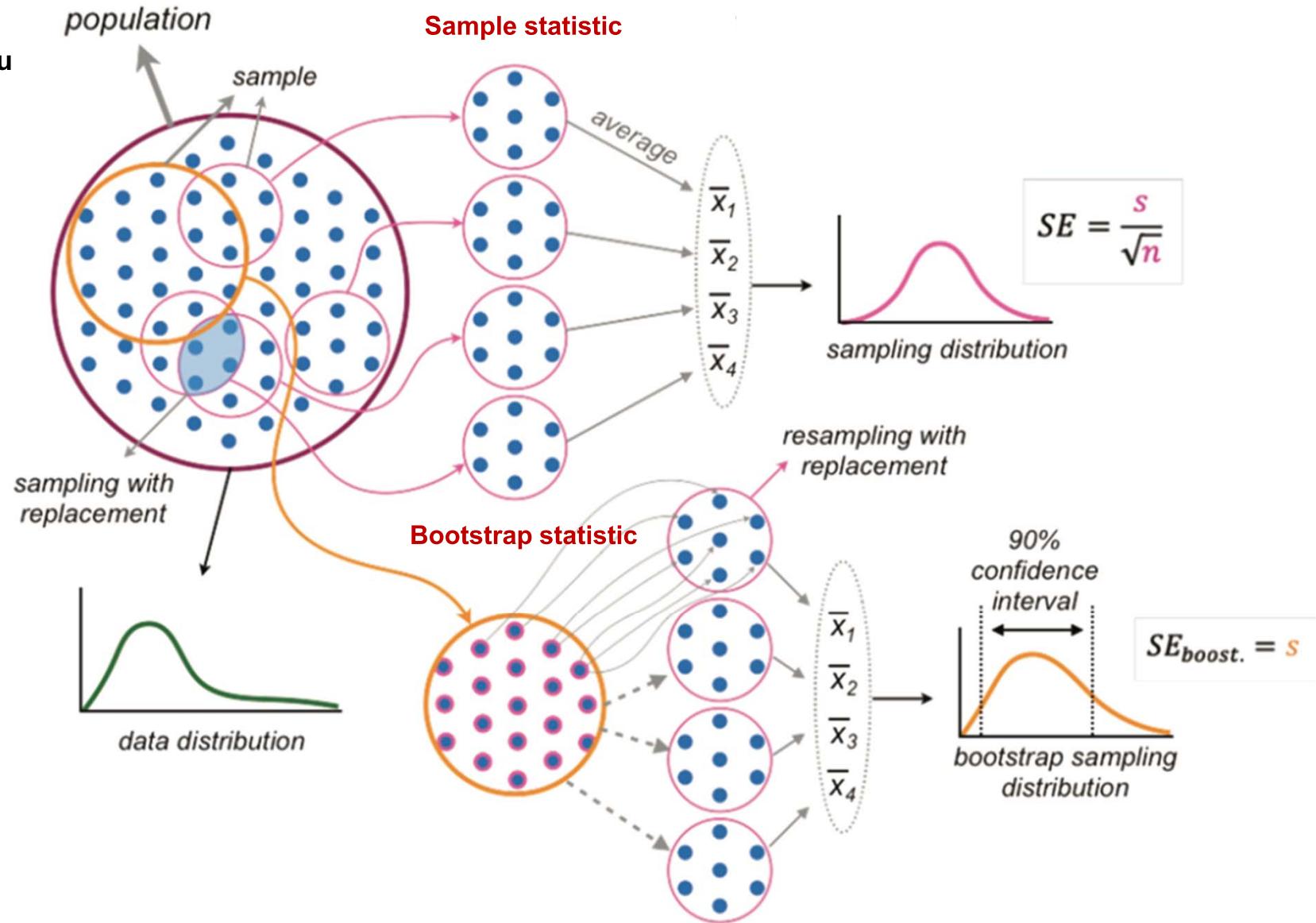
- 모집단(**population**) : 조사의 대상이 되는 모든 개체 값의 집합
- 모수(**population parameter**) : 모집단을 조사하여 얻을 수 있는 통계적 특성치
- 표본(**sample**) : 샘플링된 모집단의 부분집합

부트스트랩 알고리즘의

- 첫 단계는 원래 가지고 있는 샘플로부터 리샘플링 (복원 추출)해서 얻은 데 이터 샘플의 평균을 계산하는 것임. 유한한 데이터로부터 (보통 100개 이상의 표본이 있다고 가정합니다) 여러 번의 복원 추출을 시행하게 되니 추출된 샘플들의 평균은 조금씩 다른 결과가 나오게 됨.
- 두 번째 단계는 부트스트랩 샘플들의 평균들을 정렬하고 주어진 신뢰수준에 따른 추정 범위를 계산하는 것임. 즉, 부트스트랩 평균들로부터 주어진 백분율만큼 (신뢰수준) 평균으로부터 양쪽으로 떨어진 신뢰구간의 경계값, $\mu-, \mu+$ 를 계산하는 것임

Bootstrap

<https://www.kaggle.com/discussions/general/370664>



Bootstrap

<https://hongl.tistory.com/306>

```
import random

def resample(S, size=None):
    return [random.choice(S) for i in range(size or len(S))]

def bootstrap(x, confidence=0.68, nsamples=100):
    """Computes the bootstrap errors of the input list."""

    def mean(S):
        return float(sum(x for x in S)) / len(S)

    means = [mean(resample(x)) for k in range(nsamples)]
    means.sort()
    left_tail = int((1.0 - confidence) / 2) * nsamples
    right_tail = nsamples - 1 - left_tail
    return means[left_tail], mean(x), means[right_tail]

S = [random.gauss(2, 1) for k in range(100)]
bootstrap(S)
```

(1.755803075472523, 1.8851052326863424, 2.0112951084938935)



S의 데이터 (`random.gauss(2,1)`)는 평균이 2이고 표준편차가 1이기 때문에 μ 는 2에 가까울 것으로 기대할 수 있고 실제 결과로도 1.88의 결과를 얻음. 따라서 68%의 확률 (신뢰수준)으로 이 수들의 참 평균은 1.755과 2.011 사이에 있음을 알 수 있음. 또한, 불확실성($\mu_+ - \mu_-$)/2=0.127는 실제 표준 오차인 $\sigma/\sqrt{100}=0.1$ 과 비슷한 것을 볼 수 있음.

따라서 가지고 있는 표본으로부터 bootstrapping을 통한 모집단 분포 추정이 실제 모집단과 굉장히 근사함을 알 수 있고 부트스트랩 반복 횟수 (`bootstrap` 함수의 `nsamples` 인자)를 늘린다면 더 정확한 신뢰구간 추정이 가능

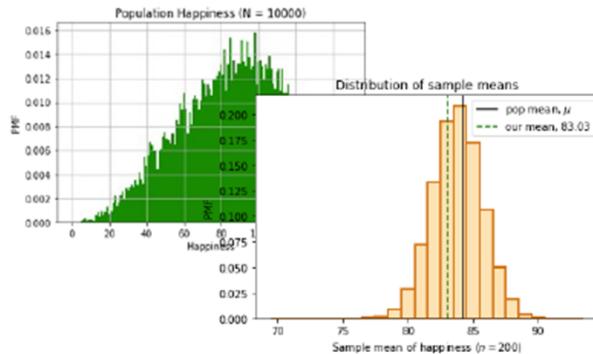
`random.choice()`

- 지정된 sequence(리스트 등) 내의 범주에서 무작위로 1개를 선택하여 추출
- for 문과 함께 사용할 경우, 중복을 허용한 N번의 무작위 추출 실행

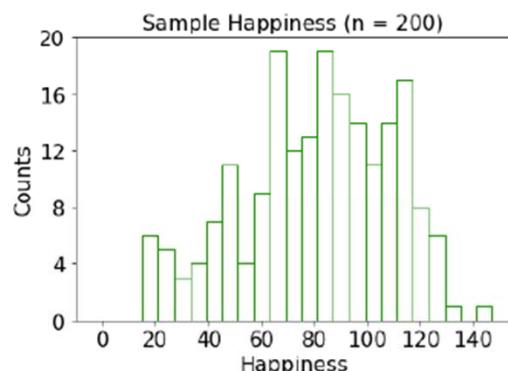
Computing statistic of sample mean

What is the standard deviation of the sample mean \bar{X} ? (sample size $n = 200$)

Population distribution
(we don't have this)



Sample distribution
(we do have this)
yay!



$$\frac{\sigma}{\sqrt{n}} = 1.886$$

1.869

$$S = \sqrt{S^2} \quad \ddot{\cup}$$

$$SE = \frac{S}{\sqrt{n}} = 1.992$$

???

Exact statistic
(we don't have this)

Simulated statistic
(we don't have this)

these are the types of things available when we know the full population

Estimated statistic,
by formula,
standard error

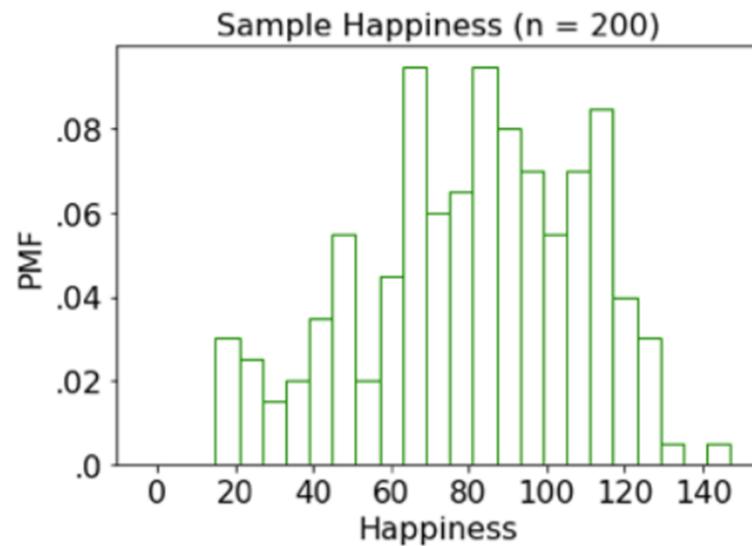
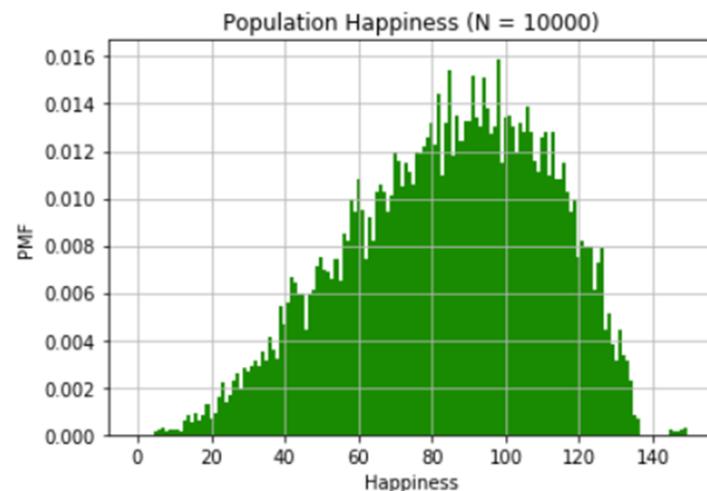
Simulated
estimated statistic

Note: We don't have access to the population.
But Doris is sharing the exact statistic with you.

© 2023 MECO54968, Mehran Sahami, and Jerry Cain, CS109, Winter 2023

Bootstrap insight 1: Estimate the true distribution

You can estimate the PMF of the underlying distribution, using your sample.*



The underlying
distribution



$$F \approx \hat{F}$$



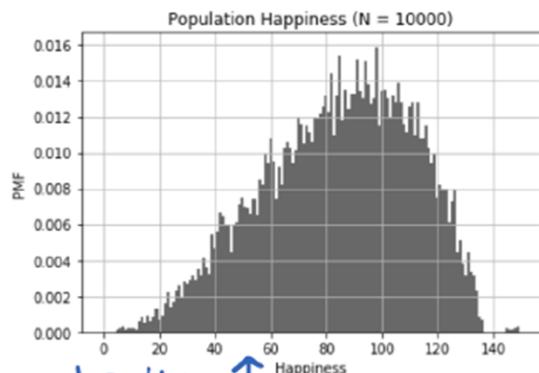
the sample distribution
(aka the histogram of
your data)
We interpret this as
a PMF

*This is just a histogram of your data!

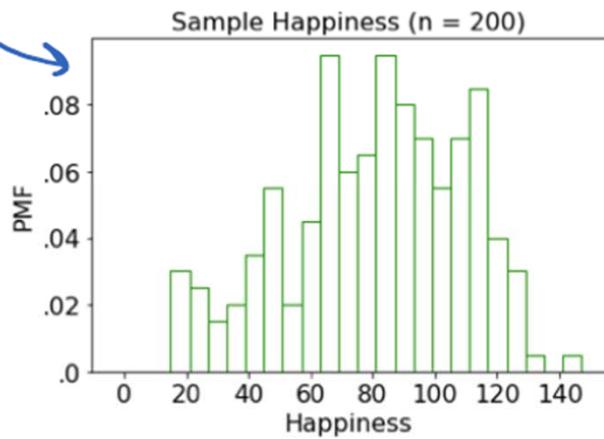
Bootstrap insight 2: Simulate a distribution

Approximate the procedure of simulating a distribution of a statistic, e.g., \bar{X} .

Population distribution
(we don't have this)



Sample distribution
(we do have this)



\approx this estimates this ↑

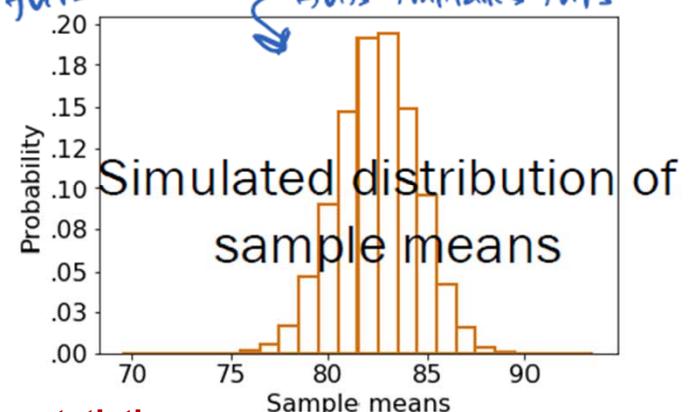
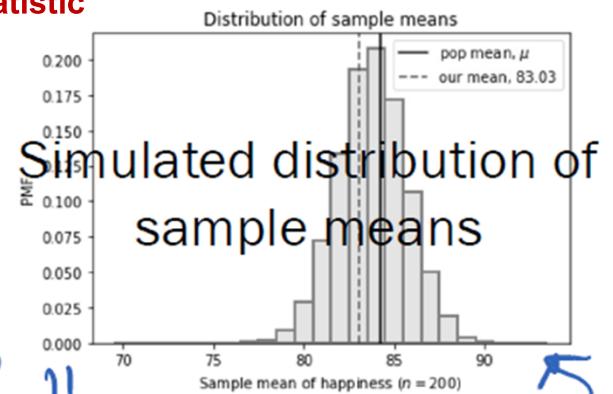
Distribution of \bar{X}

we're about to learn this!!

Bootstrap means

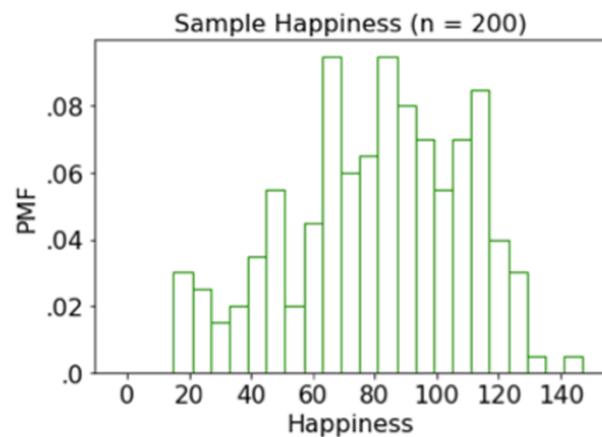
Bootstrap statistic

Sample statistic

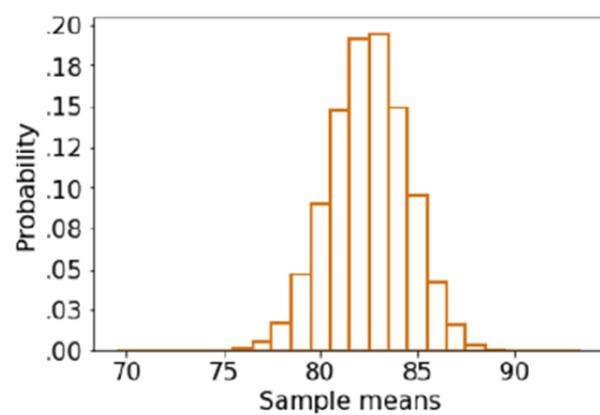


Bootstrapped sample means

```
means = [84.7, 83.9, 80.6, 79.8, 90.3, ..., 85.2]
```



Estimate the true PMF using our "PMF" (histogram) of our sample.



...generate a whole bunch of sample means of this estimated distribution...

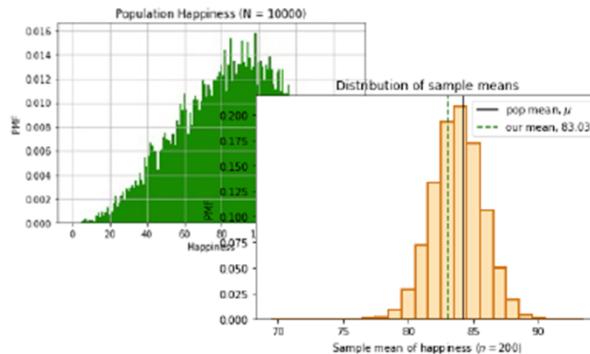
Bootstrap Standard Error
`np.std(means)`
2.003

...and compute the standard deviation of this distribution.

Computing statistic of sample mean

What is the standard deviation of the sample mean \bar{X} ? (sample size $n = 200$)

Population distribution
(we don't have this)

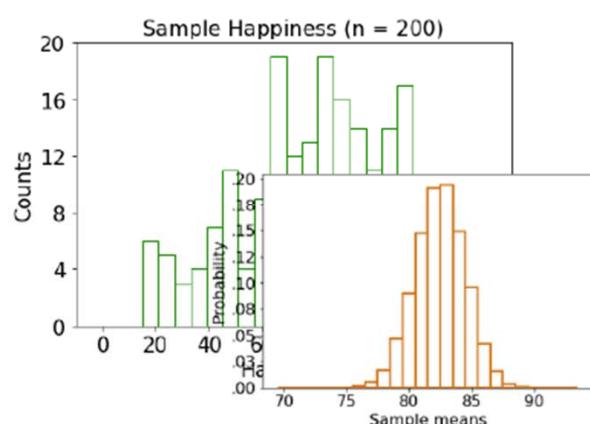


$$\frac{\sigma}{\sqrt{n}} = 1.886$$

1.869

Exact statistic
(we don't have this)

Sample distribution
(we do have this)



$$SE = \frac{S}{\sqrt{n}} = 1.992$$

✓
2.003

Simulated statistic
(we don't have this)

Estimated statistic,
by formula,
standard error

Simulated estimated
statistic, **bootstrapped**
standard error

Bootstrap algorithm

Bootstrap Algorithm (sample):

1. Estimate the PMF using the sample
2. Repeat 10,000 times:
 - a. Resample `sample.size()` from PMF
 - b. Recalculate the `sample mean` on the resample
3. You now have a distribution of your `sample mean`

[we sample with replacement, because the sample operates as a PMF.
(also, you'd just regenerate the sample every time)]

What is the distribution of your `sample mean`?

We'll talk about this algorithm in detail with a demo!

Bootstrap algorithm

Bootstrap Algorithm (sample):

1. Estimate the PMF using the sample
2. Repeat **10,000** times:
 - a. Resample `sample.size()` from PMF
 - b. Recalculate the **statistic** on the resample
3. You now have a **distribution of your statistic**

takeaway from this slide? the bootstrap algorithm is general, and not just for sample means.

What is the distribution of your **statistic**?

Bootstrapped sample variance

Bootstrap Algorithm (sample):

1. Estimate the PMF using the sample
2. Repeat 10,000 times:
 - a. Resample `sample.size()` from PMF
 - b. Recalculate the `sample variance` on the resample
3. You now have a distribution of your `sample variance`

this is computing the
bootstrapped standard
error of the
sample variance, S^2 .

What is the distribution of your `sample variance`?

Even if we don't have a closed form equation,
we estimate statistics of sample variance with bootstrapping!

Bootstrap: Sample variance

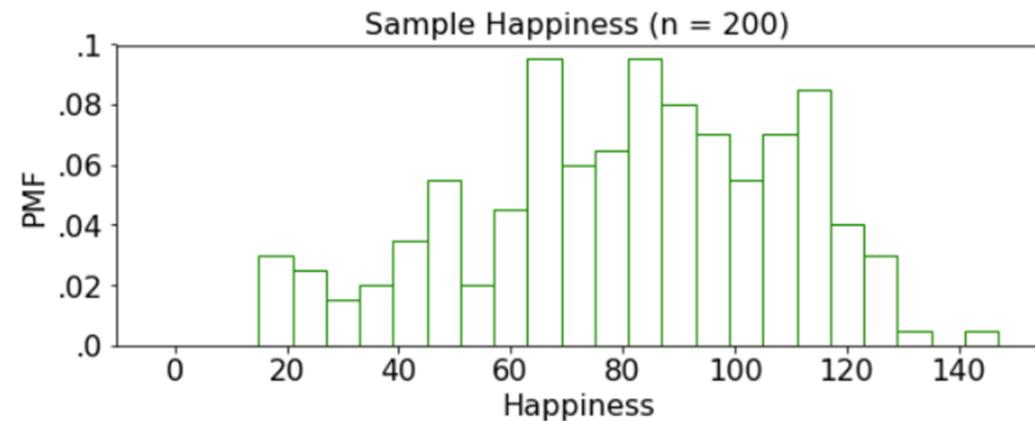
Bootstrapped sample variance

Bootstrap Algorithm (sample):

1. Estimate the PMF using the sample
2. Repeat **10,000** times:
 - a. Resample `sample.size()` from PMF
 - b. Recalculate the **sample variance** on the resample
3. You now have a **distribution of your sample variance**

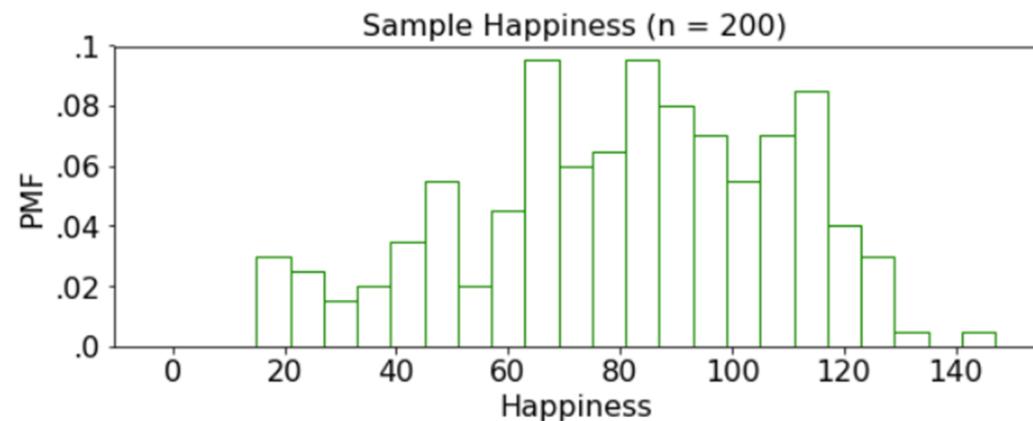
Goal What is the distribution of your **sample variance?**

Bootstrapped variance



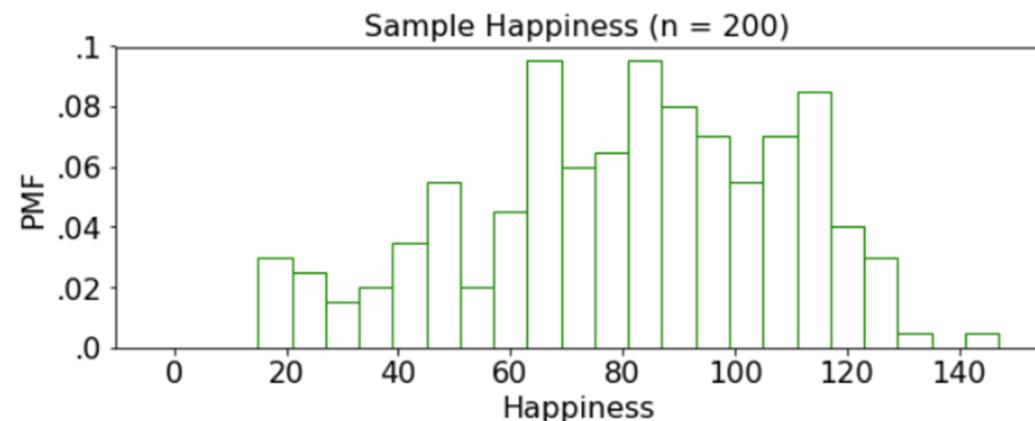
- 1. Estimate the **PMF** using the sample
- 2. Repeat **10,000** times:
 - a. Resample `sample.size()` from PMF
 - b. Recalculate the **sample variance** on the resample
- 3. You now have a **distribution of your sample variance**

Bootstrapped variance



1. Estimate the PMF using the sample
2. Repeat **10,000** times:
 - a. Resample `sample.size()` from PMF
 - b. Recalculate the `sample variance` on the resample
3. You now have a distribution of your sample variance

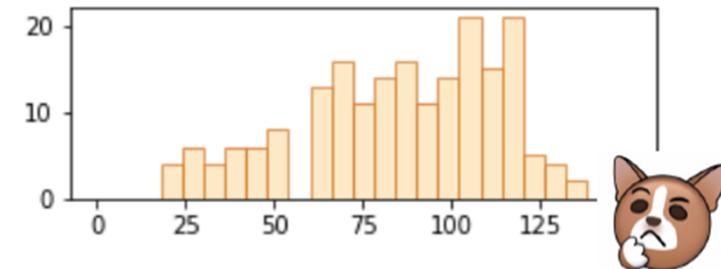
Bootstrapped variance



$x_1 \quad x_2 \quad x_3 \quad x_4 \quad \dots \quad x_{200}$

↓ ↓ ↓ ↓ ↓

[52, 38, 98, 107, ..., 94]



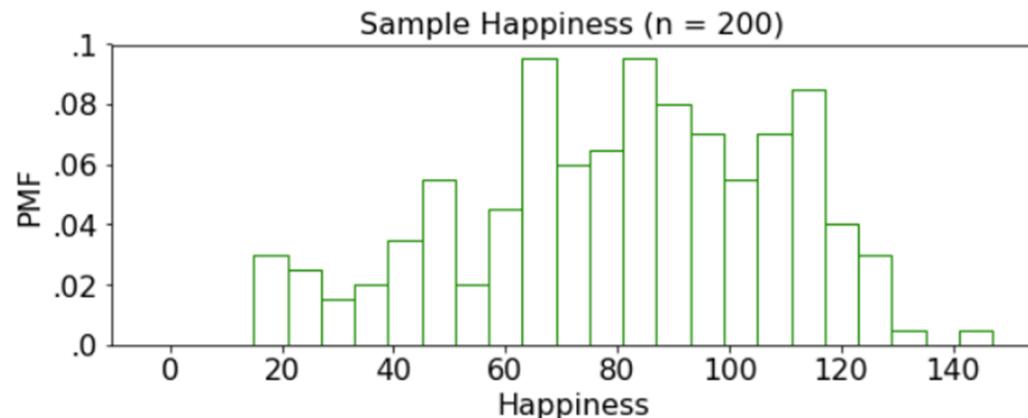
1. Estimate the PMF using the sample
2. Repeat **10,000** times: $n=200$
 - a. Resample `sample.size()` from PMF
 - b. Recalculate the **sample variance** on the resample
3. You now have a **distribution** of your



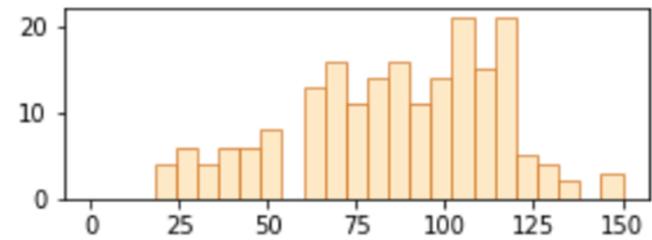
Why are these samples different?

This resampled sample is generated with replacement.

Bootstrapped variance



[52, 38, 98, 107, ..., 94]

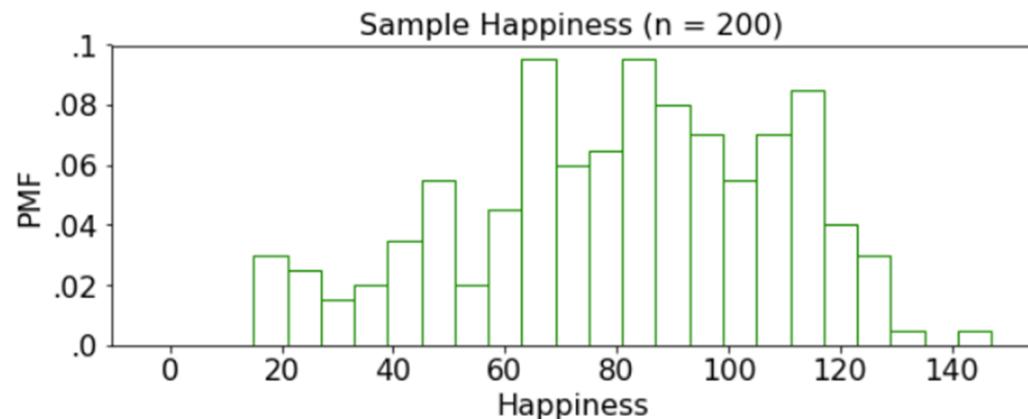


formula for S^2 is
still $\frac{1}{n-1} \sum_{k=1}^n (x_i - \bar{x})^2$

1. Estimate the PMF using the sample
2. Repeat **10,000** times:
 - a. Resample `sample.size()` from PMF
 - b. Recalculate the **sample variance** on the resample
3. You now have a **distribution** of your **sample variance**

variances = [827.4]

Bootstrapped variance

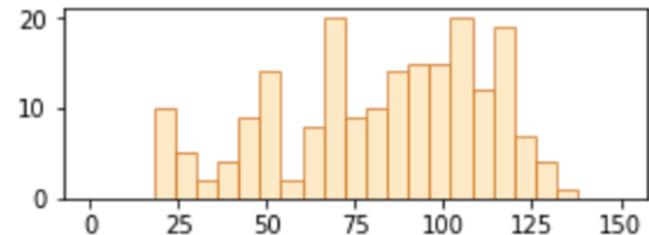
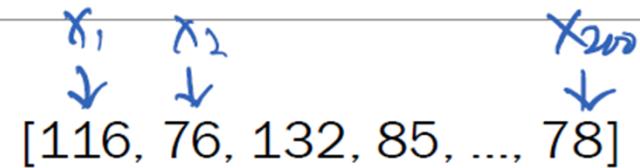
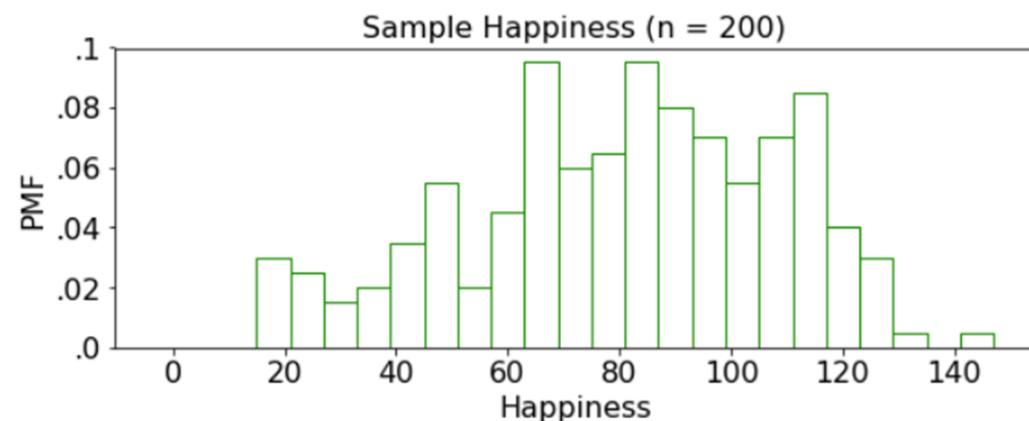


1. Estimate the PMF using the sample
2. Repeat **10,000** times:
 - a. Resample `sample.size()` from PMF
 - b. Recalculate the `sample variance` on the resample
3. You now have a `distribution of your sample variance`

`variances = [827.4]`

Bootstrapped variance

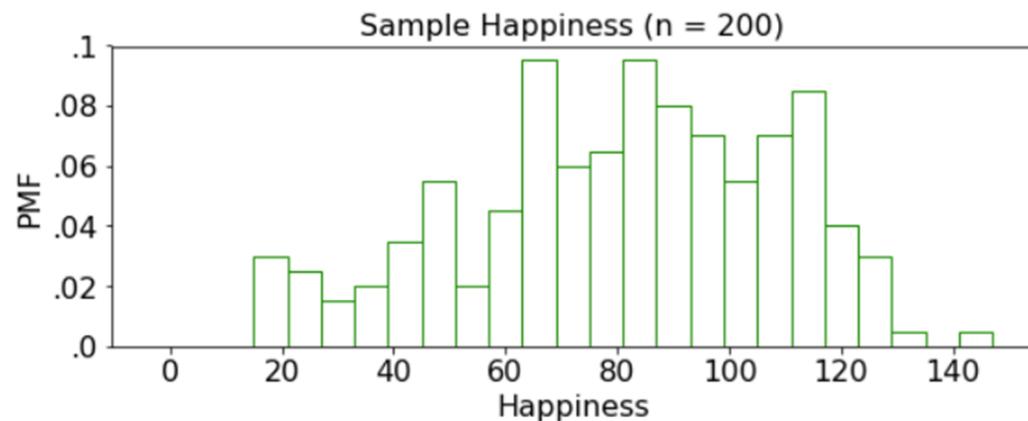
Second iteration might generate:



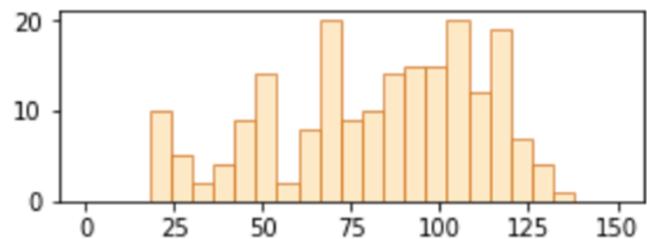
1. Estimate the PMF using the sample
2. Repeat **10,000** times:
 - a. Resample `sample.size()` from PMF
 - b. Recalculate the **sample variance** on the resample
3. You now have a **distribution of your sample variance**

variances = [827.4]

Bootstrapped variance



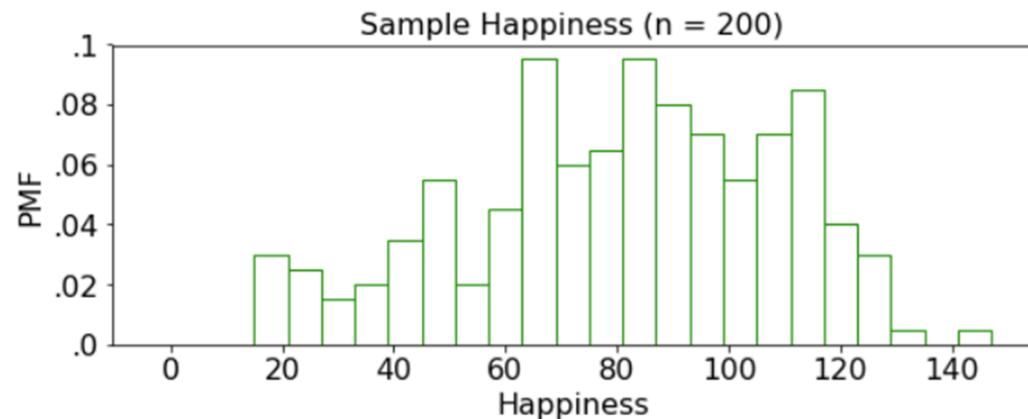
[116, 76, 132, 85, ..., 78]



1. Estimate the PMF using the sample
2. Repeat **10,000** times:
 - a. Resample `sample.size()` from PMF
 - b. Recalculate the **sample variance** on the resample
3. You now have a distribution of your sample variance

variances = [827.4, 846.1]

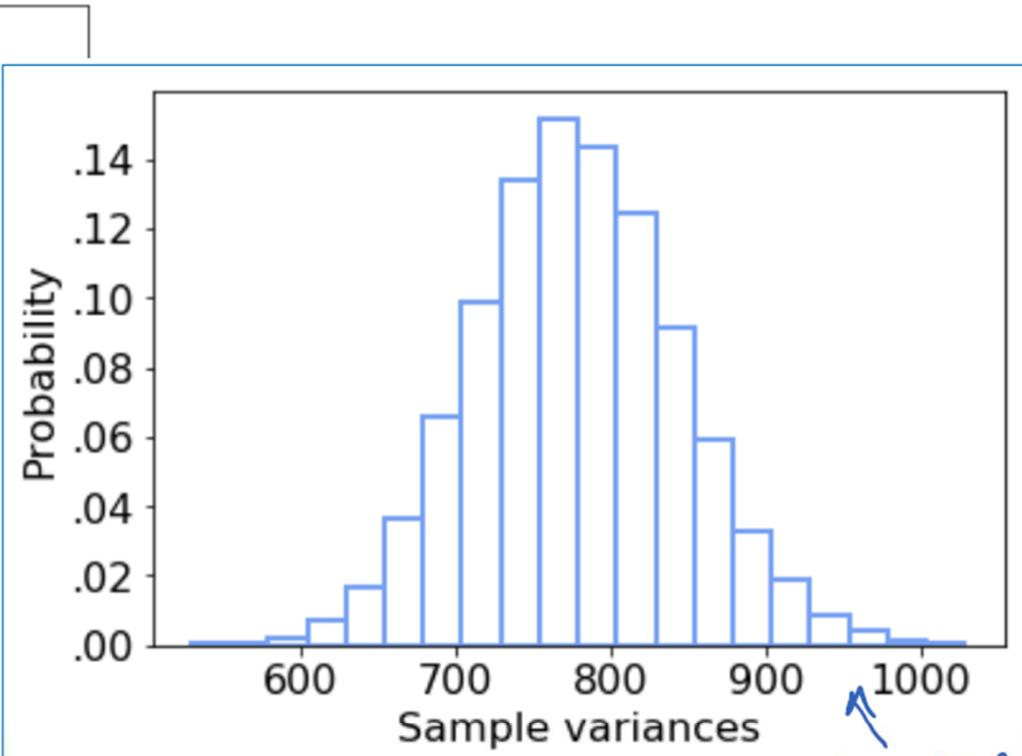
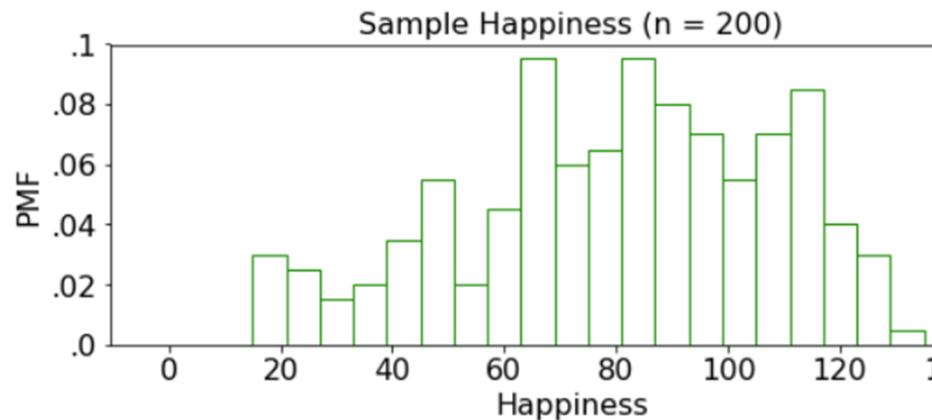
Bootstrapped variance



1. Estimate the PMF using the sample
2. Repeat **10,000** times:
 - a. Resample `sample.size()` from PMF
 - b. Recalculate the **sample variance** on the resample
3. You now have a distribution of your sample variance

variances = [827.4, 846.1]

Bootstrapped variance



1. Estimate the PMF using the :
2. Repeat **10,000** times:
 - a. Resample `sample.size()`
 - b. Recalculate the `sample`
3. You now have a distribution of your sample variance
`variances = [827.4, 846.1, 726.0, ..., 860.7]`

this is a graph of this

Bootstrapped variance

3. You now have a distribution of your sample variance

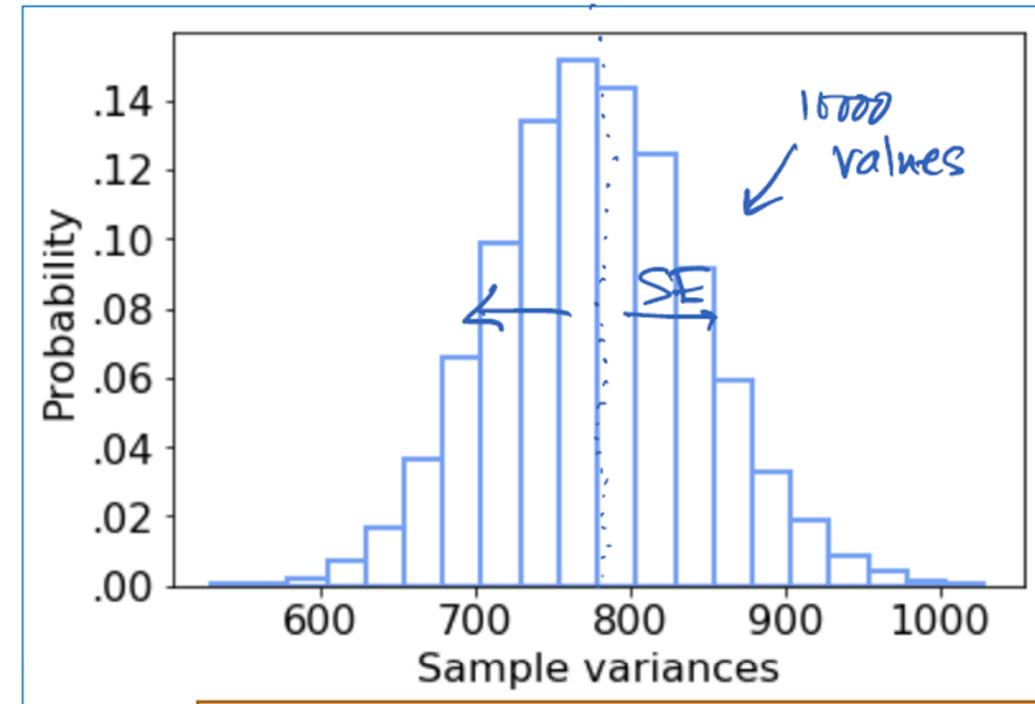
`variances = [827.4, 846.1, 726.0, ..., 860.7]`

of length 10000

What is the bootstrapped standard error?

`np.std(variances)`

Bootstrapped standard error: 66.16



- Simulate a distribution of sample variances
- Compute standard deviation

Standard error

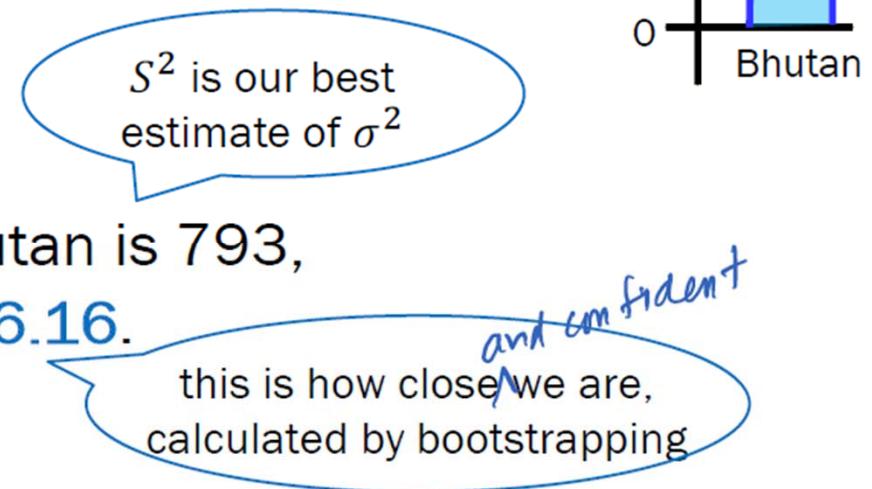
1. Mean happiness:

Claim: The average happiness of Bhutan is 83, with a standard error of 1.99.

Closed form: $SE = \sqrt{\frac{S^2}{n}}$

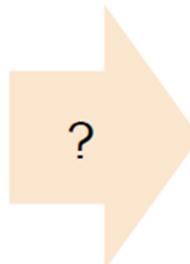
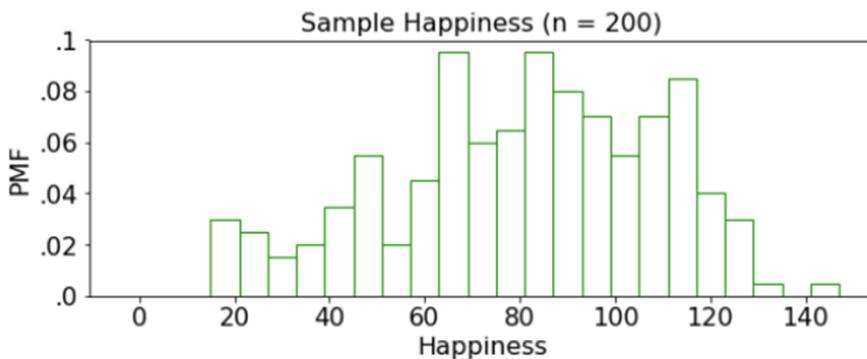
2. Variance of happiness:

Claim: The variance of happiness of Bhutan is 793, with a **bootstrapped standard error of 66.16**.

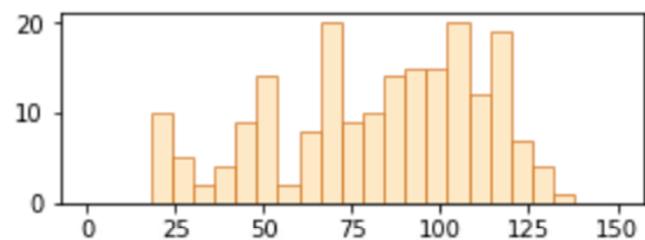


Algorithm in practice: Resampling

1. Estimate the PMF using the sample
2. Repeat 10,000 times:
 - a. Resample `sample.size()` from PMF
 - b. Recalculate the statistic on the resample
3. You now have a distribution of your statistic



[116, 76, 132, 85, ..., 78]



$$P(X = k) = \frac{\text{\# values in sample equal to } k}{n}$$

Algorithm in practice: Resampling

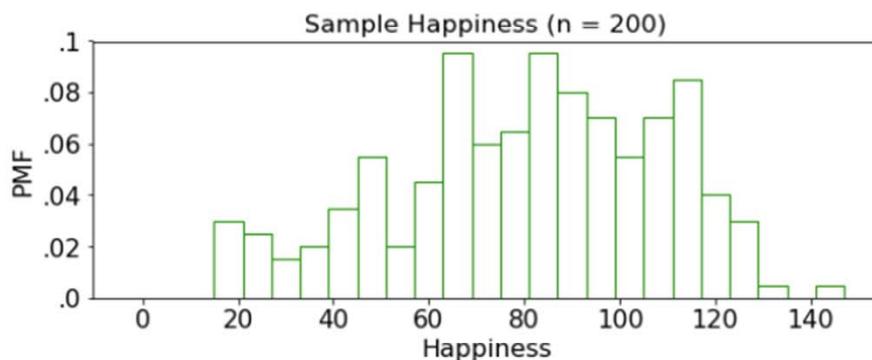
```
def resample(sample, n):
    # estimate the PMF using the sample
    # draw n new samples from the PMF ✓
    return np.random.choice(sample, n, replace=True)
```

size : int or tuple of ints, optional

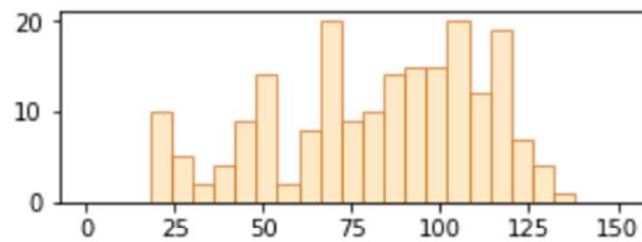
Output shape. If the given shape is, e.g., (m, n, k), then m * n * k samples are drawn. Default is None, in which case a single value is returned.

replace : boolean, optional

Whether the sample is with or without replacement. Default is True, meaning that a value of a can be selected multiple times.



[116, 76, 132, 85, ..., 78]



$$P(X = k) = \frac{\text{\# values in sample equal to } k}{n}$$

This resampled sample is generated with replacement.

To the code!

Bootstrap provides a way to calculate probabilities of statistics using code.

Bootstrapping works for any statistic*

*as long as your sample is iid and the underlying distribution does not have a long tail

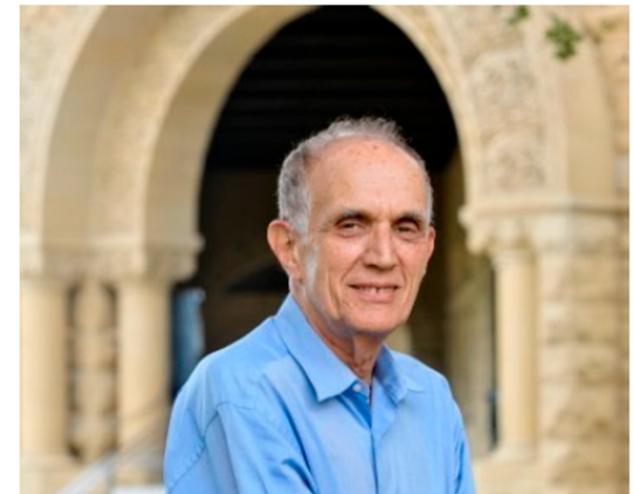
fancy way of saying that variance
is finite
Google colab notebook [link](#)

[Sampling and Bootstrap.ipynb - Colaboratory \(google.com\)](#)

<https://colab.research.google.com/drive/18q4oXxJUiFSrUzT15YuiGox32RFBuno?usp=sharing#scrollTo=G0f60GZBoR7h>

Bradley Efron

- Invented bootstrapping in 1979
- Still a professor at Stanford
- Won a National Science Medal



Inventor of Efron's dice: 4 dice A, B, C, D where:

$$P(A > B) = P(B > C) = P(C > D) = P(D > A) = \frac{2}{3}$$



Bootstrap: p-value



[참고] <https://kongdols-room.tistory.com/153>

○ Null hypothesis (귀무가설, 영가설) : 모집단 확률변수와 추정치가 차이가 없다는 가설이다.

▷ 수학적 기호로 H_0 라고 표현한다.

○ Alternative hypothesis (대안가설, 대립가설) : 모집단 확률변수와 추정치가 차이가 있다는 가설이다.

▷ 수학적 기호로 H_A 라고 표현한다.

○ 검정을 수행하기 위한 p-value(관찰된 유의수준)

○ p-value은 Null hypothesis이 참일 때 주어진 데이터 집합을 얻을 수 있는 확률이다.

○ p-value은 관찰된 데이터가 Null hypothesis의 신뢰성을 0에서 1 사이의 수치로 표현한 것이다.

- p-value가 작을수록 신뢰성이 낮다고 본다.
- 특정 값(대개 0.05나 0.01 등)보다 작을 경우 Null hypothesis를 Reject하는 것이 관례이나 항상 그렇지는 않다.

○ 데이터 집합으로부터 계산된 p-values는 Null hypothesis의 신뢰성을 측정할 때 사용될 수 있다.

양방향 t 검정(Two side t-test)

양방향 구간에 대해 다음과 같이 귀무가설이 검정될 수 있다.

단방향 t 검정(one side t-test)

단방향 구간에 대해 다음과 같이

확률변수 X 는 위에서 양방향 t-검정에서 계산한 방식과 동일하게 계산된다.

$$H_0: \mu = \mu_0 \quad \text{versus} \quad H_A: \mu \neq \mu_0$$

$$p\text{-value} = 2 \times P(X \geq |t|), \quad X = \frac{\bar{x} - \mu_0}{s/\sqrt{n}}$$

$$H_0: \mu \leq \mu_0 \quad \text{versus} \quad H_A: \mu > \mu_0$$

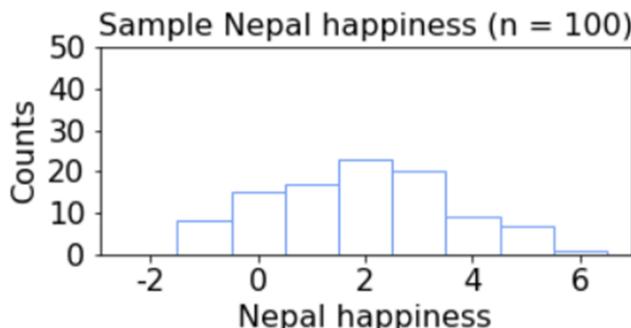
$$p\text{-value} = P(X \geq t), \quad X = \frac{\bar{x} - \mu_0}{s/\sqrt{n}}$$

$$H_0: \mu \geq \mu_0 \quad \text{versus} \quad H_A: \mu < \mu_0$$

$$p\text{-value} = P(X \leq t), \quad X = \frac{\bar{x} - \mu_0}{s/\sqrt{n}}$$

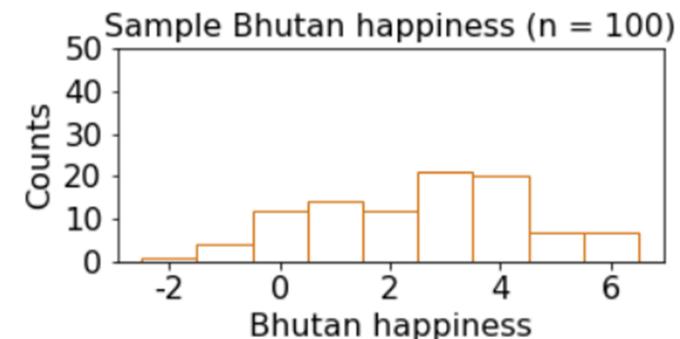
Null hypothesis test

Nepal Happiness
4.45
2.45
6.37
2.07
...
1.63



$$\bar{X}_1 = 3.1$$

Bhutan Happiness
0.91
0.34
1.91
1.61
...
1.08



$$\bar{X}_2 = 2.4$$

Claim: The difference in mean happiness between Nepal and Bhutan is 0.7 happiness points, and **this is statistically significant**.

Null hypothesis test

def null hypothesis – Even if there is no pattern (i.e., the two samples are from identical distributions), your claim might have arisen by chance.

def p-value – What is the probability that the observed difference occurs under the null hypothesis? *that is, what are the chances you observed a significant difference by chance?*

Example:

- Flip some coin 100 times.
- Flip the same coin another 150 times.
- Compute fraction of heads in both groups.
- There is a possibility we'll see the observed difference in these fractions even if we used the same coin

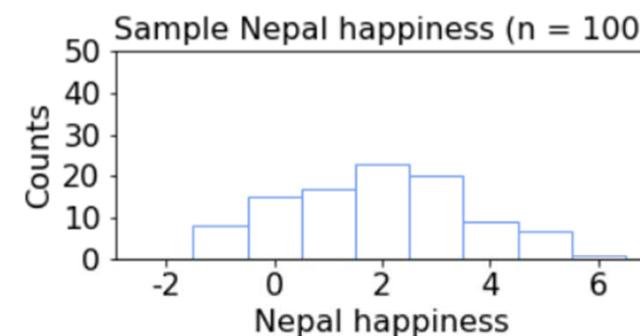
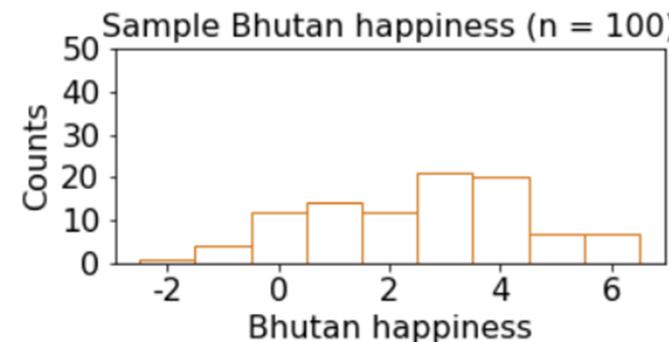
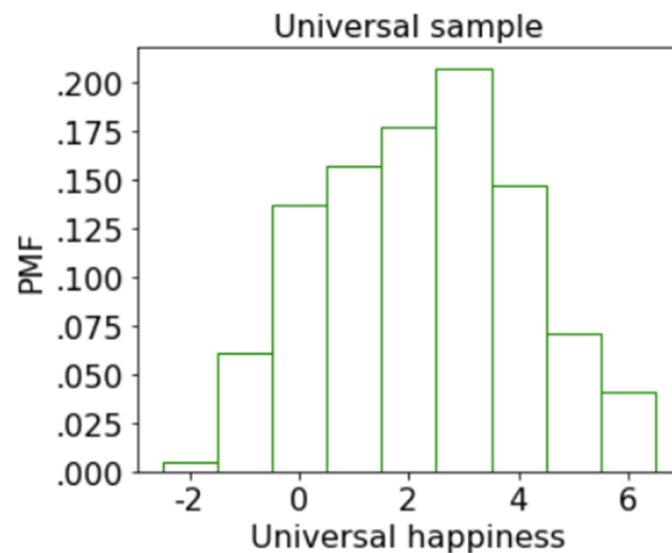
} Null hypothesis assumes we use the same coin

} p-value

A significant p-value (< 0.05) means we reject the null hypothesis.

Universal sample

(this is what the null hypothesis assumes)



$$\bar{X}_1 = 3.1$$

how often do
we see this when
drawn from
combined
population?

$$\bar{X}_2 = 2.4$$

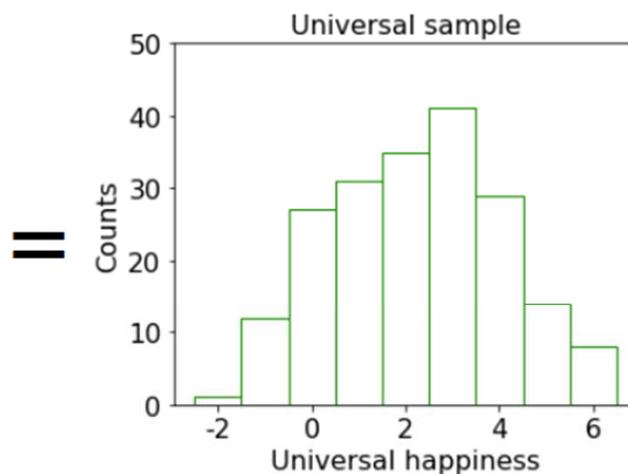
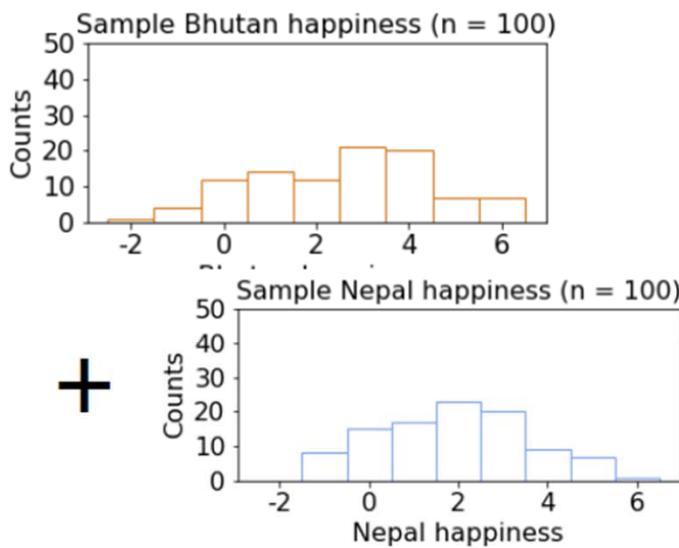
Want **p-value**: probability $|\bar{X}_1 - \bar{X}_2| = |3.1 - 2.4|$ happens under null hypothesis

Bootstrap for p-values

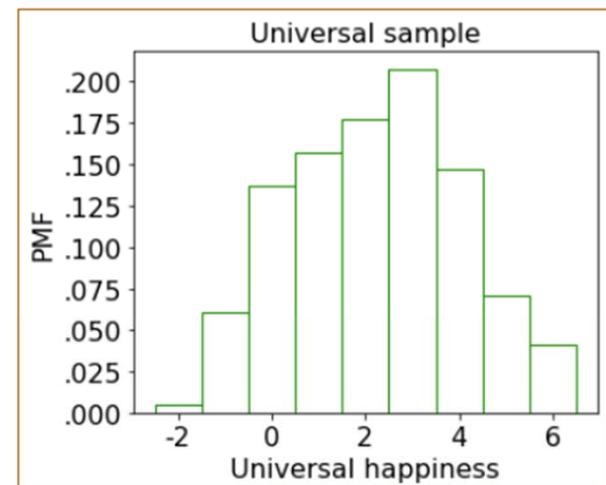
1. Create a **universal sample** using your two samples

if two statistics are different by chance,
then combining population shouldn't
matter, right?

i.e., recreate the null hypothesis

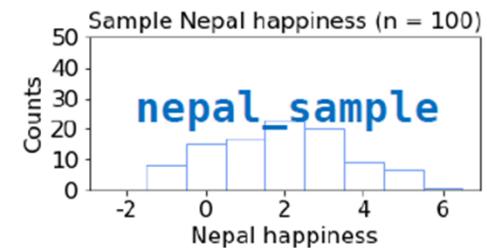
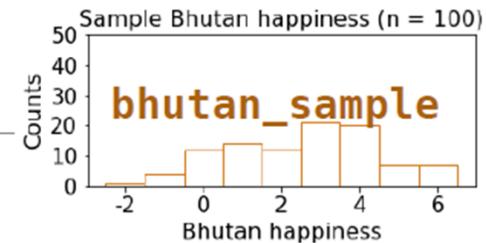


≈



Bootstrap for p-values

1. Create a **universal sample** using your two samples
2. Repeat **10,000** times:
 - a. Resample **both samples**
 - b. Recalculate the **mean difference** between the resamples
3. **p-value** = $\frac{\# (\text{mean diffs} \geq \text{observed diff})}{n = 10000}$



Probability
that observed
difference arose
by chance

Bootstrap for p-values

```
def pvalue_boot(bhutan_sample, nepal_sample):
    N = size of the bhutan_sample
    M = size of the nepal_sample
    observed_diff = |mean of bhutan_sample - mean of nepal_sample|

    uni_sample = combine bhutan_sample and nepal_sample
    count = 0

    repeat 10,000 times:
        bhutan_resample = draw N resamples from the uni_sample
        nepal_resample = draw M resamples from the uni_sample
        muBhutan = sample mean of the bhutan_resample
        muNepal = sample mean of the nepal_resample
        diff = |muNepal - muBhutan|
        if diff >= observed_diff:
            count += 1

    pValue = count / 10,000
```

Bootstrap for p-values

1. Create a universal sample using your two samples

```
def pvalue_boot(bhutan_sample, nepal_sample):
    N = size of the bhutan_sample
    M = size of the nepal_sample
    observed_diff = |mean of bhutan_sample - mean of nepal_sample|
    
    uni_sample = combine bhutan_sample and nepal_sample ← this is the backbone
    count = 0
    of a null hypothesis
    
repeat 10,000 times:
    bhutan_resample = draw N resamples from the uni_sample
    nepal_resample = draw M resamples from the uni_sample
    muBhutan = sample mean of the bhutan_resample
    muNepal = sample mean of the nepal_resample
    diff = |muNepal - muBhutan|
    if diff >= observed_diff:
        count += 1

pValue = count / 10,000
```

Bootstrap for p-values

2. a. Resample both samples

```
def pvalue_boot(bhutan_sample, nepal_sample):
    N = size of the bhutan_sample
    M = size of the nepal_sample
    observed_diff = |mean of bhutan_sample - mean of nepal_sample|

    uni_sample = combine bhutan_sample and nepal_sample
    count = 0

    repeat 10,000 times:
        bhutan_resample = draw N resamples from the uni_sample
        nepal_resample = draw M resamples from the uni_sample
        muBhutan = sample mean of the bhutan_resample
        muNepal = sample mean of the nepal_resample
        diff = |muNepal - muBhutan|
        if diff >= observed_diff:
            count += 1

    pValue = count / 10,000
```

imperative grammar
go with sizes
that are the same as those
when derived separately!
otherwise you're using
different random
variables!

Bootstrap for p-values

2. b. Recalculate the **mean difference** b/t resamples

```
def pvalue_boot(bhutan_sample, nepal_sample):
    N = size of the bhutan_sample
    M = size of the nepal_sample
    observed_diff = |mean of bhutan_sample - mean of nepal_sample|

    uni_sample = combine bhutan_sample and nepal_sample
    count = 0

    repeat 10,000 times:
        bhutan_resample = draw N resamples from the uni_sample
        nepal_resample = draw M resamples from the uni_sample
        muBhutan = sample mean of the bhutan_resample
        muNepal = sample mean of the nepal_resample
        diff = |muNepal - muBhutan|
        if diff >= observed_diff:
            count += 1

    pValue = count / 10,000
```

Lisa Yan, Chris Piech, Mehran Sahami, and Jerry Cain, CS109, Winter 2023

Stanford University 66

Bootstrap for p-values

$$3. \text{ p-value} = \frac{\# (\text{mean diffs} > \text{observed diff})}{n}$$

```
def pvalue_boot(bhutan_sample, nepal_sample):
    N = size of the bhutan_sample
    M = size of the nepal_sample
    observed_diff = |mean of bhutan_sample - mean of nepal_sample|

    uni_sample = combine bhutan_sample and nepal_sample
    count = 0

    repeat 10,000 times:
        bhutan_resample = draw N resamples from the uni_sample
        nepal_resample = draw M resamples from the uni_sample
        muBhutan = sample mean of the bhutan_resample
        muNepal = sample mean of the nepal_resample
        diff = |muNepal - muBhutan|
        if diff >= observed_diff:
            count += 1

    pValue = count / 10,000
```

Bootstrap for p-values

```
def pvalue_boot(bhutan_sample, nepal_sample):
    N = size of the bhutan_sample
    M = size of the nepal_sample
    observed_diff = |mean of bhutan_sample - mean of nepal_sample|

    uni_sample = combine bhutan_sample and nepal_sample
    count = 0

    repeat 10,000 times:
        bhutan_resample = draw N resamples from the uni_sample
        nepal_resample = draw M resamples from the uni_sample
        muBhutan = sample mean of the bhutan_resample
        muNepal = sample mean of the nepal_resample
        diff = |muNepal - muBhutan|
        if diff >= observed_diff:
            count += 1

    pValue = count / 10,000
```

with replacement!

Bootstrap



Let's try it!

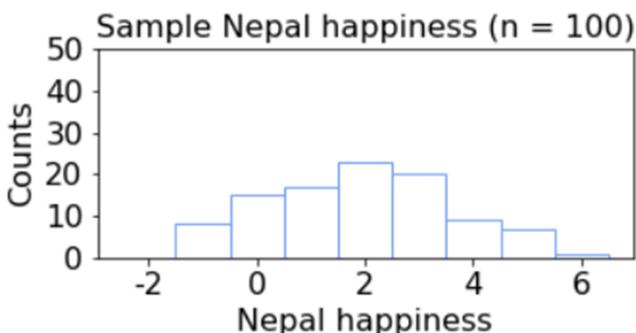
[Sampling and Bootstrap.ipynb - Colaboratory \(google.com\)](#)

<https://colab.research.google.com/drive/18q4oXxJUiFSrUzT15YuiGox32RFBuno?usp=sharing>

Google colab notebook [link](#)

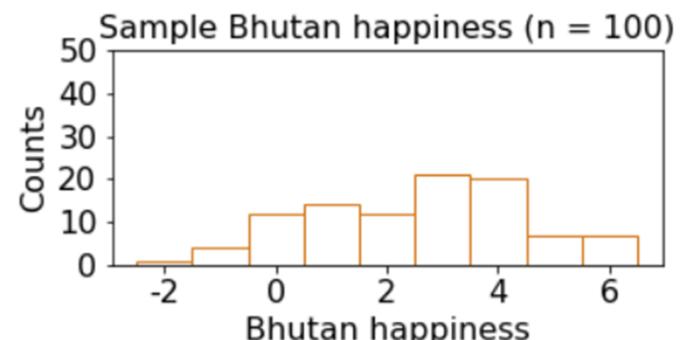
Null hypothesis test

Nepal Happiness
4.45
2.45
6.37
2.07
...
1.63



$$\bar{X}_1 = 3.1$$

Bhutan Happiness
0.91
0.34
1.91
1.61
...
1.08



$$\bar{X}_2 = 2.4$$

Claim: The happiness of Nepal and Bhutan have a 0.7 difference of means, and this is statistically significant ($p < 0.05$).